UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE MÁSTER

AUTOMATIC CLASSIFICATION OF VIDEOS USING FEATURE DESCRIPTIONS

Máster Universitario En Ingeniería De Telecomunicación

Autor: María Narváez Encinal
Tutores: Álvaro García y Tobias Senst
Ponente: José M. Martínez

Junio 2017

AUTOMATIC CLASSIFICATION OF VIDEOS USING FEATURE DESCRIPTIONS

Autor: María Narváez Encinal
Tutores: Álvaro García y Tobias Senst
Ponente: José M. Martínez

Video Processing and Understanding Lab
Computer Engineering Department
Escuela Politécnica Superior
Universidad Autónoma de Madrid

In collaboration with

Institut Telekommunikationssysteme
Fachgebiet Nachrichtenübertragung
Technische Universität Berlin

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)

Acknowledgments

I have to say that, in this case, there are millions of things to thank, not only for the support received during the development of this project but for giving me the opportunity to live one of the most incredible experiences of my life in such a wonderful city like Berlin.

First of all, I would like to thank Chema for this first proposal of collaboration with the Technical University of Berlin. To my tutor, Alvaro, for those thousands of mails whose response took little time to reach and, above all, his constant support.

Special mention for the people with whom I worked at the University of Berlin. Thomas Sikora for making my stay possible there. To Tobias, my tutor, for his explanations, for his patience, his understanding and his help in any subject outside the project. To the whole team, to make me feel like I was at home. Really, there are not enough words to thank you.

My classmates, because after all we have done this way together; Manu, Alex, Gabri ... Marta, the great discovery of these last courses, thank you for being there and for that express visit to Berlin.

On the other hand, there is always someone who supports you above everything, whatever happens, those who give you the most strength, those who send you iberian ham from Extremadura when you are out of Spain and they worry more about what you eat than what you work for. Those who blindly trust you, that you will get everything that you propose...Again, thank you mom, thank you dad, obviously without you none of this would have been possible. Andre, I do not say anything about you because you give me a lot of war...I am joking, you do not know how much I love you.

Finally, to my Berlin, what magical and incredible you are. For all that has taught me this city, the people I met there, who endured me and who continue to endure me despite the distance, thank you for being part of those 4 months so great. I WILL BE BACK!

María Narváez Encinal. June 2017.

Resumen

Los sistemas de video vigilancia son cada vez más imprescindibles en lo que a seguridad se refiere. Gracias al aumento del número de cámaras y las horas de grabación, surge la necesidad de crear sistemas de reconocimiento automático que permitan detectar diferentes situaciones o eventos que resulten significativos desde el punto de vista de la seguridad.

Este proyecto se centra en la detección automática de violencia a partir del análisis único de las imágenes (frames) del vídeo, es decir, eliminando información adicional como puede ser audio o información de contexto. Actualmente, existen varios algoritmos desarrollados pensando en este tipo de aplicación, los más destacados: MoSIFT, LaSIFT o STIP. Todos ellos se basan en la extracción y descripción de características.

El algoritmo propuesto, LaSP-SIFT, se basa en la descripción de la apariencia y del movimiento del video mediante el descriptor SP-SIFT. La extracción del movimiento se realiza mediante el cálculo de las trayectorias de cada uno de los píxeles de la imagen durante un número de frames determinado. A estas trayectorias se las conoce como medidas de Lagrangian. Una vez descrito tanto la apariencia como el movimiento de todos los frames, se aplica la técnica Bolsa de Palabras (BoW) para extraer los descriptores más representativos del conjunto completo, también denominados vocabulario o diccionario. Hecho esto, se calcula un único histograma de frecuencias por vídeo mediante el vocabulario extraído anteriormente y dichos histogramas serán la entrada al entrenamiento de la Máquina de Vector Soporte final. Finalmente se evaluarán los clasificadores entrenados y se compararán con los algoritmos del estado del arte.

Palabras clave

Descripción de características, medidas de Lagrangian, Bolsa de Palabras, Máquina de Soporte Vectorial.

Abstract

Video surveillance systems are becoming more and more indispensable as far as security is concerned. Thanks to the increase in the number of cameras and recording times, the need arises to create automatic recognition systems that allow to detect different situations or events that are significant from the point of view of security.

This project focuses on the automatic detection of violence based on the unique analysis of the images (frames) of the video, that is, removing additional information such as audio or context information. Currently, there are several algorithms developed with this type of application in mind: MoSIFT, LaSIFT or STIP. All of them are based on the extraction and description of characteristics.

The proposed algorithm, LaSP-SIFT, is based on the description of the appearance and movement of the video using the SP-SIFT descriptor. The extraction of the movement is done by calculating the paths of each of the pixels of the image during a certain number of frames. These trajectories are known as Lagrangian measurements. After describing both the appearance and the movement of all frames, the Bag-of-Words (BoW) technique is applied to extract the most representative descriptors of the complete set also called vocabulary or dictionary. Once this is done, a single frequency histogram per video is calculated using the vocabulary extracted previously and these histograms will be the input to the training of the final Vector Support Machine. Finally we evaluate the trained classifiers and compare them with the state of the art algorithms.

Key words

Features description, Lagrangian measures, Bag-of-Words, Support Vector Machine.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation.

Video surveillance systems have been gaining importance over the years being strange not to find them in football stadiums, airports, public transport stations, etc. The continuous improvement of the technologies along with the increase of the use of the computer vision techniques has made efforts focus on creating security systems able to recognize certain events of automatic form. The emergence of these automatic systems and their evolution allow to reduce the staff in charge of controlling and visualizing the security videos and in addition, it allow to increase the number of cameras without worrying about the monitoring of these. With this we will have more complete systems.

The actions recognition using computer vision techniques has acquired a great importance thanks to these automatic video surveillance systems. There is a wide variety of algorithms implemented in security cameras capable of detecting different events without human supervision. This area of work presents many difficulties due to the large number of factors involved such as crowded places, lighting changes, shadows, video quality, etc. In addition, human actions are associated with a highly random spatio-temporal behavior, that is, the realization of the same action is never identical to another, which makes it difficult to detect.

Within the actions recognition this work will be focused on classify video sequences as violent or not violent. In order to achieve this goal will be necessary to use diverse techniques of image processing like extraction and description of points of interest or the obtaining an estimation that describe the motion between different frames. On the other hand, the concept of machine learning will be used to obtain a classifier capable of labeling a new entry video as violent or non-violent.

1.2 Objetives.

The main objetive of this project is the desing, implementation and test of an algorithm that would be able to classify a video sequence like a violent or nor violent. To achieve this goal will be necessary to train a vector support machine that has the ability to differentiate between violent or non-violent scenes. To train these support vector machine we are going to describe video sequences using appearance and motion information of each frame following different algorithms.

In order to achieve the final classifier, we can break down the objectives as follows:

1. **Detailed study of the state of art.** This study can be divided in four stages; on the first one, we will review different forms of violent events classification that use data sources such as audio, context information and video. On the second one, we will study the appearance description techniques, this is detection and description of interest points. Then we will study the motion estimation and description. And finally, we will analyzes the use of a support vector machine to classify video sequence and the advantages of a Bag-of-Words model.

2. **Study and understanding of the SP-SIFT code developed in the VPULab.**

3. **Code implementation.** The code development can be structured as follows: video description using appearance descriptors, extraction and description of the direction Lagrangian measures, calculation of the vocabulary of the Bag-of-Words model and description of each video with only one histogram using the previous vocabulary and, finally, train and test a Support Vector Machine (SVM) that classify a new video sequence as violent or non-violent.

4. **Evaluation of results.** We will comparatively evaluate the results of different techniques of description of appearance and movement. The performance of Lagrangian measures with different integration times will be evaluated as well as the influence of the number of clusters or words of the vocabulary of the Bag-of-Words model.

1.3 Memory structure.

The memory of the work is divided into the following chapters:

- Chapter 1. Motivation, project objetives and memory structure.

- Chapter 2. In this chapter, State of the art, we explain some of the current techniques studied that will help us to understand the algorithm proposed.

- Chapter 3. We describe in detail the approach developed, explaining the possible advantages and disadvantages of each stage implemented.
- Chapter 4. Here we explain the evaluation method and we compare our algorithm with the state of the art algorithms.
- Chapter 5. Conclusions derived from the analysis of the results and enumeration of possible future work lines.
- References and Appendices.

Chapter 2

State of the art

This chapter provides an overview of previous work in the areas related to the project objective. With the following sections we understand the project structure and the main steps to achieve the final goal: classify video sequences as violent or non-violent.

2.1 Introduction

At present, there are multiple proposals focused on violent events detection. They can be classified depending on the type of data source that use to achieve the goal: audio and video, only audio or only video. There are even algorithms that use context information, such as subtitles, for violent movie classification.

Some of proposals for the violence recognition use video and audio data, one of the first approaches was introduced by Nam *et al.* [1] and uses flame and blood detection combined with the degree of motion and the characteristic sounds of violent event to recognize violent scenes in videos. More recently, to detect violence in movies Gong *et al.* [2] proposes the use of high-level audio effects, low-level visual and auditory features to characterize the shots with fast-tempo. Ling and Wang [3] split the detection step into two views: from audio-view, a weakly-supervised method is exploited and from the video-view, they use a classifier to detect shots by using blood, flame and explosion detection and motion estimation. Finally, both classifiers are combined in a co-training way. Giannakopoulos *et al.* [4] uses the statistics of audio features and average and orientation variance of motion combined in a k-Nearest Neighbor classifier to decide if the sequence is or not violent.

There are also methods that only use audio data to detect violent events, for example, Cheng *et al.* [5] recognize shots, explosions and car-breaking in audio through the use of Gaussian mixture (GMM) and Hidden Markov models (HMM) and Gian-

nakopoulos *et al.* [6] also propose audio features to classify violence content.

In surveillance systems typically do not exist audio or context information so appear methods capable to detect violence just through visual analysis. Clarin *et al.*[7] detects skin and blood pixels and analyzes motion intensity to classify violent actions but the main disadvantage of this method is that only detects violent actions involving blood. There are global descriptors such a ViF (Violent Flows) [8] that represents the statistics of flow-vector magnitudes dynamics over time. Déniz *et al.* [9] proposes a method based on the acceleration of global motion getting reduce the run time of the previous algorithms. On the other hand, Nievas *et al.* [10] compares two different bag-of-words models using local spatio-temporal point descriptions: the STIP feature and MoSIFT. Space-Time Interest Point (STIP), explained in [11], detect the interest points at multiple spatial and temporal scales, then histograms of oriented gradients (HOG) [12], histograms of optical flow (HOF) [12] and a combination of the previous histograms are extracted to describe these interest points. MoSIFT [13] is an extension of SIFT descriptor [14] that consists in a standard SIFT descriptor to describe the appearance and an analogous histogram of optical flow to describe the motion estimation. MoSIFT is significantly more computationally expensive than STIP but gets better performance. Xu *et al.* [15] improves the performance of MoSIFT using a sparse coding instead of bag-of-words models to detect violence in videos. These approaches only consider two frames to estimate motion information and sometimes we need to take into account larger temporal periods to be able to describes some actions. On a microscopic level, exists several algorithms that use space time volumes of stacked silhouettes to recognize actions [16]. To also describe large temporal periods Wang *et al.* [17] proposed to use long term trajectories created by tracking either feature points or densely sampled points.

Nowadays, the Lagrangian theory has gained importance in the video analysis. Senst *et al.* [18] propose a local feature, called LaSIFT, based on SIFT descriptor to encode the direction Lagrangian measures for violent event detection. Based on LaSIFT and SP-SIFT features descriptor we propose a new bag-of-words model capable of detecting violent sequences by describing local features and the trajectories of these local features through SP-SIFT, our approach will be dercribed in Section 3.

The following section describes in detail the main methods of describing features used to detect violence in video sequences.

2.2 Features description

To be able to classify videos as violent or non-violent it is necessary to describe them previously and, according to that description, the classifier will make a decision. Approaches based on feature descriptors for events classification typically do not only use appearance description of frames but also describe motion information. In the following subsections we will explain some of the most important descriptors.

2.2.1 Scale-Invariant Feature Transform

The Scale-Invariant Feature Transform (SIFT) method proposed by Lowe [14] is able to describe the appearance of interest points using a fixed-dimension vector. This method can be divided in two stages: interest points detection and description. The detection stage is to extract points that, due to their characteristics, are susceptible to remain the same despite the changes in the image. That is, find those points that are invariant to typical transformations such as translations, rotations, changes of scale, perspective or illumination. When the interest points have been detected it is necessary to describe them. The aim of the description stage is to obtain, for each point, a distinctive and relevant information of the surrounding region, getting an invariant description of the typical transformations enumerated above.

SIFT detector

The SIFT detection of keypoints candidates is carried out as follows:

1. Obtaining the Scale-Space, $L(x, y, \sigma)$, by covolution of the original image, $I(x, y)$, with a Gaussian filter with an increasing scale factor, σ. This is

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.1}$$

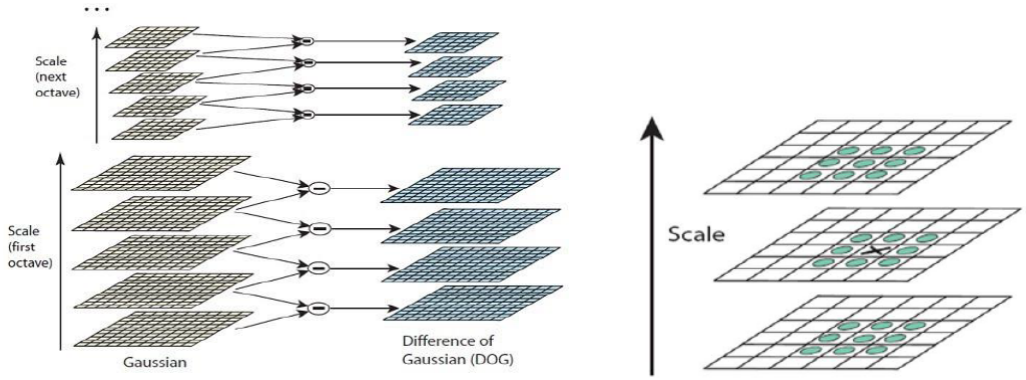
   , where $G(x, y, \sigma)$ is the Gaussian filter defined as

$$G(x, y, \sigma) = \frac{1}{2 \cdot \pi \cdot \sigma} e^{-(x^2+y^2)/2\sigma} \tag{2.2}$$

2. Gaussian Laplacian approximation. This is done by calculating the difference-of-Gaussian of consecutive scales (DoG):

$$D(x, y, \sigma) = L(x, y, k \cdot \sigma) - L(x, y, \sigma) \tag{2.3}$$

   Steps one and two are made for different image sizes generating the concept



(a) Gaussian pyramid concept. (b) Max and min location

Figure 2.1: SIFT detector.

of the Gaussian Pyramid. The Gaussian Pyramid is obtained by rescaling the last smoothed image from the previous octave. This process can be seen in the Figure 2.1a.

3. Detection of blobs, that is, local maxima and minima. As shown in Figure 2.1b, in order to detect the local maxima and minima of $D(x, y, \sigma)$, each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below. For each max or min found, output is the location and the scale.

Once the keypoints candidates are located, the SIFT algorithm eliminates those that have poor stability. In the first place, SIFT eliminates, through a threshold, those interest points that have low contrast. For achieve stability, to reject keypoints with low contrast is not sufficient. The difference-of-Gaussian function will have a strong response along edges, even if the location along the edge is poorly determined and therefore unstable to small amounts of noise. These type of candidates should also be eliminated and for this the Hessian matrix, H, is used. The Hessian matrix contain the second derivatives of the defference-of-Gaussian function in the scale where is the keypoint candidate:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.4}$$

$$Tr\,(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{2.5}$$

$$Det\,(H) = D_{xx} \cdot D_{yy} - (D_{xy})^2 = \alpha \cdot \beta \tag{2.6}$$

It can be demostrated that the points that do not comply with the Equation 2.7 are instability so they will be discarded.

$$\frac{Tr\,(H)^2}{Det\,(H)} < \frac{(r+1)^2}{r} \tag{2.7}$$

Lowe in [14] uses a value of r equal to 10.

Thanks to the use of the concept of Gaussian Pyramids, it is possible to approximate the Gaussian Laplacian with a lower computational cost and, in addition, to get the interest points invariant to scale.

SIFT descriptor

To calculate the SIFT descriptor we must know the scale where the keypoint has been detected and work on the smoothed image, in this way the descriptor will be invariant to scale. The first step to represent each interest point is obtaining the gradient of the smoothed image and its orientation. Once this is done, we calculate, through Equation 2.8 and 2.9, the main local orientation and the magnitude of the gradient for each interest point. If we describe each point respect to its main orientation, we will get it to be invariant to rotations.

$$\theta(x,y) = \arctan \frac{L\,(x,y+1) - L\,(x,y-1)}{L\,(x+1,y) - L\,(x-1,y)} \tag{2.8}$$

$$m\,(x,y) = \sqrt{(L\,(x+1,y) - L\,(x-1,y))^2 + (L\,(x,y+1) - L\,(x,y-1))^2} \tag{2.9}$$

A 36-bins orientation histogram is generated (1 orientation per $360^{\text{o}}/36$ bins = 10^{o}) on which each pixel in the local region votes as a function of the magnitude (weighted with a Gaussian filter centered on the point to be described) and orientation of its rotated gradient respect to the dominant interest point orientation calculated in Equation 2.8. The largest peak in the histogram will indicate the main orientation of the keypoint. There may be several major addresses (above 80%), these are used to create duplicate interest points in the same position and scale but with different orientation.

Now a characteristic vector containing a local statistic of the gradient orientations is calculated. Following the previous approach and as can be seen in Figure 2.2

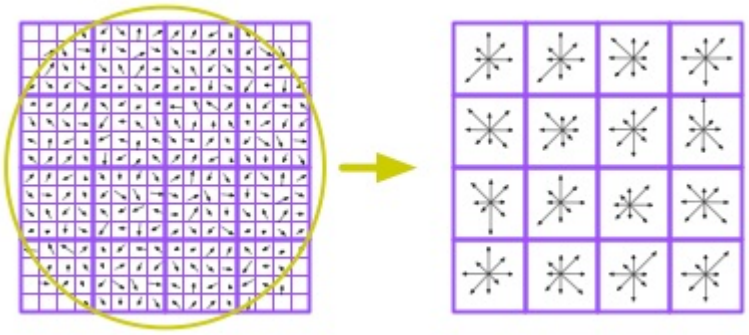


Figure 2.2: SIFT descriptor.

a region of 16x16 pixels around the point of interest is described. This region is divided into subregions of 4x4 pixels obtaining a histogram of orientations of 8 bins according to [14] (1 orientation per $360^{o}/8$ bins = 45^{o}) for each subregion, in this way it is achieved that the descriptor is robust to local displacements. The descriptor is extracted from the concatenation of the 16 histograms generated by the previous subregions obtaining a 128-dimensional vector and, finally, this characteristic vector is normalized to provide it a certain robustness against illumination changes.

2.2.2 Motion SIFT

The Motion SIFT or MoSIFT algorithm [13] is an extention of the well-known SIFT image descriptor. As we have explained in Subsection 2.2.1, the standard SIFT extracts histogram of oriented gradients in the image and build a 128-dimensional descriptor.

The MoSIFT algorithm detects spatially distinctive interest points tith substantial motions, how we can be seen in Figure 2.3, Chen *et al.* [13] first apply the SIFT detection algorithm to find visually distintive components in the spatial domain and then detect spatio-tempotal interest points with motion constraints. The motion constraint consists of a 'sufficient' amount of optical flow around the distinctive points. To extract the motion information is necessary obtain the optical flow between two frames so MoSIFT calculate optical flow pyramids over two Gaussian pyramids and a local extreme from DoG pyramids can only become an interest point if it has sufficient motion in the optical flow pyramid.

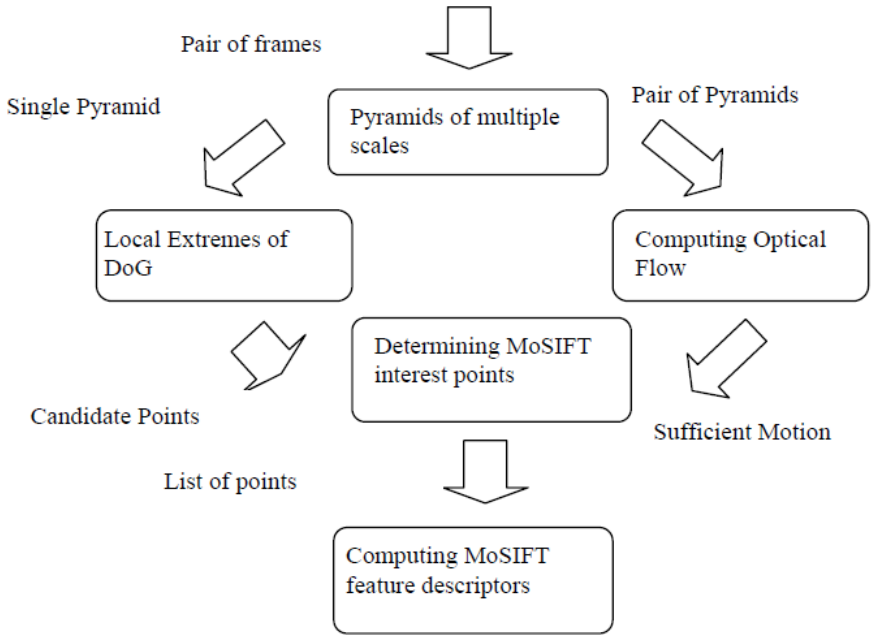


Figure 2.3: MoSIFT detection.

Finally, the 256-dimensional MoSIFT descriptor is designed to represent the interest point in two parts. The first part is an aggregated histogram of gradients (HoG) to describe the spatial appearance such as SIFT so the first 128 dimensions of characteristic vector are the standard SIFT image descriptor. The second part is an aggregated histogram of optical flow (HoF) which represent local motion so the remaining 128 dimensions of the MoSIFT descriptor are an analogous histogram of optical flows adapting the idea of grid aggregation in SIFT to describe motions. Optical flow detects the magnitude and direction of a movement, thus has the same properties appearance gradients and can be treated in the same way.

2.2.3 Space-Time Interest Point

As described in [11], Space-Time Interest Points (STIP) is an extension of the Harris corner detection operator to space-time. The idea is to extend the notion of Harris interest points in the spatial domain by requiring the image values in local spatio-temporal volumes to have large variations along both the spatial and the temporal directions. Points with such properties will correspond to spatial interest points with distinct locations in time corresponding to local spatio-temporal neighborhoods with non-constant motion.

We remember that the main idea of the Harris corner detector is to find regions

with changes of intensity in multiple directions. In smooth areas of the image there will be no variation in the intensity, in the case of a contour, the intensity variation will occur only in one direction and, in the case of a corner, the variation in intensity will be seen in multiple directions. To locate these zones the Harris detector performs a multiscale analysis through the Laplacian-Gaussian space. We define the Harris matrix such as:

$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.10}$$

where I_x and I_y are the partial derivatives of I. This matrix must have two big eigenvalues to be a point of interest. The calculation of the eigenvalues has a high computational cost, therefore the trace and the determinant of the matrix are used to establish a corners map, R (Equation 2.13).

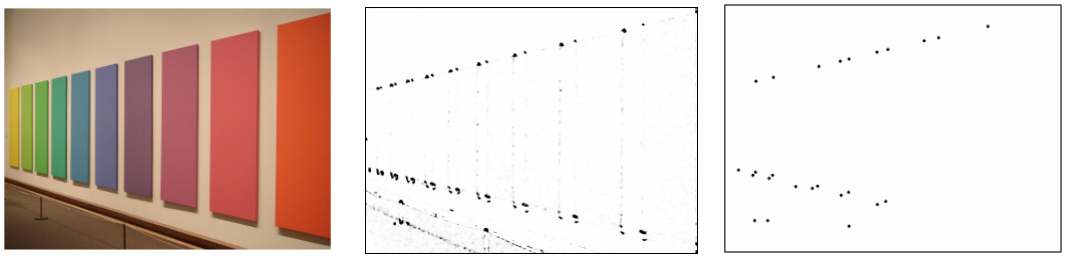
$$Tr\,(A) = I_x^2 + I_y^2 \tag{2.11}$$

$$Det\,(A) = I_x^2 I_y^2 - (I_x I_y)^2 \tag{2.12}$$

$$R = Det\,(A) - k{\cdot}Tr\,(A)^2 \tag{2.13}$$

where k value is a arbitrary constant. The original article [19] use $k = 0.04$. Finally it is necessary to discard several of the points obtained previously, to this stage it is known as non-maximal suppression. The first step is to define a threshold for the R function above a certain value, and thus discard several of the pixels that are marked as possible corners. To avoid multiple detections in the same corner, a filter called non-maximal suppression is used. This filter eliminates all points in which the direction of the gradient is not the maximum in a local environment.

STIP detected interest points will not only be characterized by a high variation of the intensity in the space but also by the non-constant motion in time. Similar to the spatial domain, we consider a spatio-temporal second-moment matrix, which is a 3-by-3 matrix composed of first order spatial and temporal derivatives averaged using a Gaussian weighting function then Laptev *et al.* [11] calculate the eigenvalues and decide if the point is a keypoint.



(a) Input image. (b) Corners map, R. (c) Non-maximal suppression filter.

Figure 2.4: Example Harris corter detector.

2.2.4 Violent Flows

The main challenge of this approach is to be able to detect violent events in real time in crowded scenarios, Hassner *et al.* [8]consider statistics of how flow-vector magnitudes change over time to detect violent scenes in video surveillance. These statistics, collected for short frame sequences, are represented using the VIolent Flows (ViF) descriptor and these descriptors are then classified as either violent or non-violent using linear Support Vector Machine (SVM). So given a sequence of frames, they produce the VIolence Flows (ViF) descriptor by first estimating the optical flow between pairs of consecutive frames. This provides for each pixel, $p(x,y,t)$, a flow vector, $(u(x,y,t), v(x,u,t))$, that matching it to a pixel in the next frame $t+1$ and they consider only the magnitude of these vector $m(x,y,t) = \sqrt{u(x,y,t)^2 + v(x,y,t)^2}$.

Unlike previous methods, do not consider the magnitudes themselves, but how they change over time. This is because although flow vectors encode meaningful temporal information, their magnitudes depend on frame resolution, different motions in different spatio-temporal locations, etc, so they are arbitrary quantities. By comparing magnitudes we obtain meaningful measures of the significance of observed motion magnitudes in each frame compared to its predecessor so they calculate a binary indicator, $b(x,y,t)$, with the following expression:

$$b(x,y,t) = \begin{cases} 1 & |m(x,y,t) - m(x,y,t-1)| \geq th \\ 0 & otherwise \end{cases} \tag{2.14}$$

where th is a threshold adaptively set in each frame to the average value of $|m(x,y,t) - m(x,y,t-1)|$. With this we get a map that reflects the significance of the magnitude change between frames t and $t+1$. They next compute a mean magnitude-change map by simply averaging these binary values, for each pixel, over

all the frames:

$$\bar{b}(x,y) = \frac{1}{T}\sum_{t} b(x,y,t) \tag{2.15}$$

The ViF descriptor is therefore produced by partitioning $\bar{b}$ into MxN non-overlapping cells and collecting magnitude change frequencies in each cell separately. The distribution of magnitude changes in each such cell is represented by a fixed-size histogram. These histograms are then concatenated into a single descriptor vector to get a description of the whole video sequence.

2.3 Bag-of-Words model

Originally, the Bag-of-Words (BoW) model was used for text processing allowing modeling documents based on word dictionaries. In the computer vision field this idea is used to represent video sequences or images as vectors of visual words occurrence.

In video sequence classification, the main advantage of this approach is its computational efficiency being able to represents each video sequence as a histogram over a set of visual words to generate a fixed-dimensional encoding that can be processed using a standard classifier. In a learning phase, the vocabulary of visual words is typically defined as the cluster centers obtained from some clustering algorithm over a large collection of sample keypoint descriptors (SIFT, SP-SIFT...).

The following steps explain in detail the Bag-of-Words model applied to computer vision [20]. This steps can be seeing in Figure 2.5.

1. Region detection. The first step of the BoW methodology is to detect local interest regions or points.

2. Interest points description.

3. Quantification of descriptors in words to form visual vocabulary. When the keypoints are detected and their features are extracted, such as with the SIFT descriptor, the final step of extracting the BoW feature from images is based on vector quantization. In general, the k-means clustering algorithm is used for this task, and the number of visual words generated is based on the number of clusters.

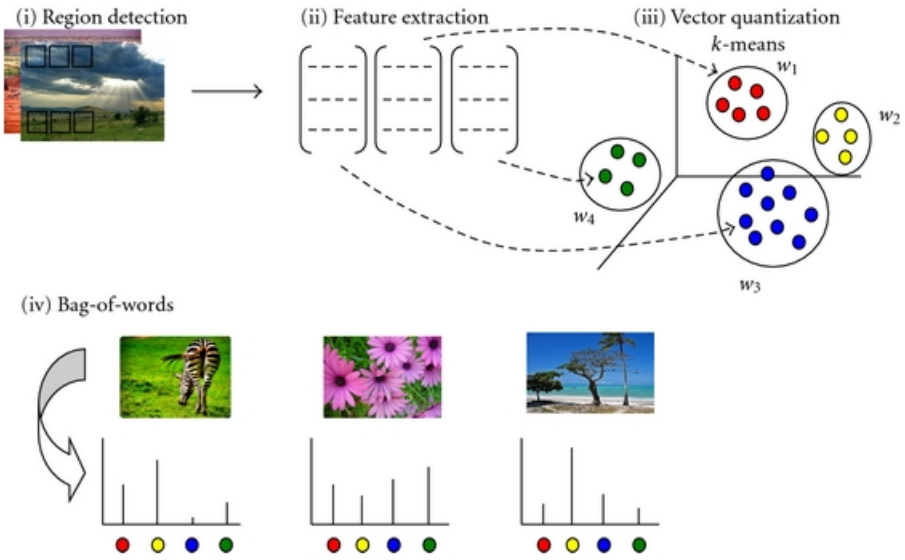


Figure 2.5: Bag-of-Word model.

4. Calculation of frequency histograms. With all image descriptors and visual words calculated by k-means, an histogram of occurrence is obtained by minimizing the distance between descriptors and visual words.

2.3.1 Clustering

As explained above, the vocabulary of visual words is achieved by some clustering algorithm. These types of algorithms are based on the grouping of vectors according to certain criteria, usually as a function of distance or similarity. We will explain below one of the most used clustering algorithms: k-means.

k-means

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data. The goal of this algorithm is to find groups in the data. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

The algorithm inputs are the number of clusters K and the data set. The data set is a collection of features for each data point, in computer vision this will be the features vector extracted in the description step of the point of interest. The algorithms starts with initial estimates for the K centroids, each centroids defines a clusters, and then iterates between two steps to achieve split the data set in groups:

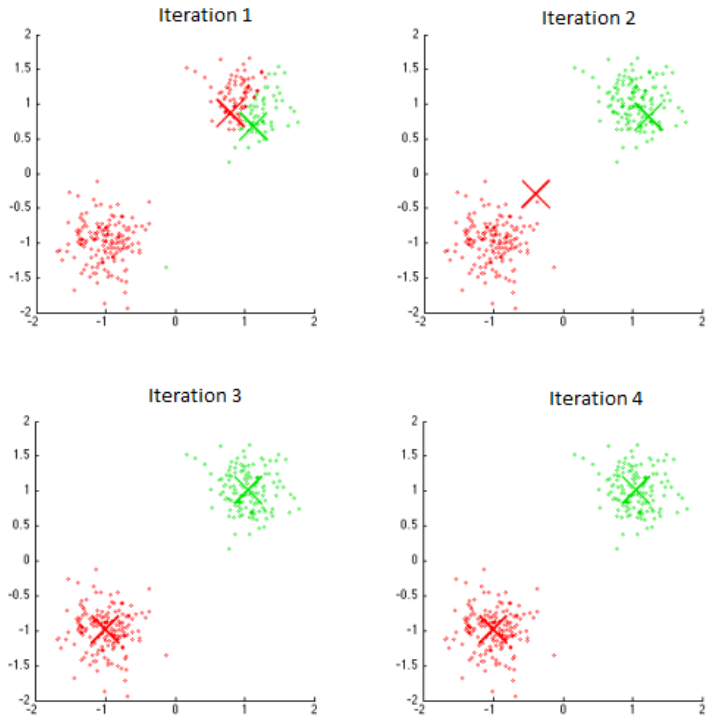


Figure 2.6: K-means algorithm example.

1. Data assigment: The distance between each data and the existing centroids is calculated. Each data is assigned the closest centroid based on the squared Eucliden distance.

2. Centroid update: When all points in the data set have been assigned to the nearest centroid, it is necessary to recalculate the centroids. This is done by taking the mean of all data points assigned to that centroid's cluster.

The algorithm iterates between steps one and two until a stopping criteria is met, for example, the centroids no change. In the Figure 2.6, we can see an example with 4 iterations, in the fifth iteration there will be no variation of the centroids and, therefore, the algorithm will end.

2.4 Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. They are considered linear learning

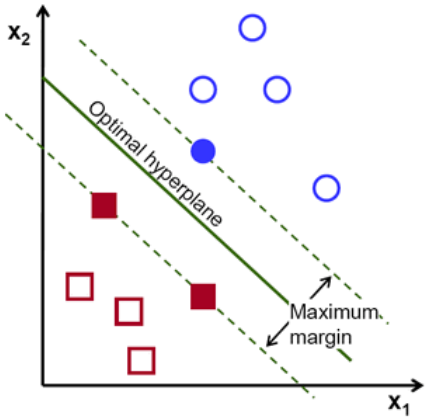


Figure 2.7: Example 2-dimensional SVM.

machines and like any supervised learning algorithm, a previous training is necessary before the classifier can automatically classify a new input data. One of the main advantages is that SVMs are effective in high dimensional spaces, fot this reason they are widely used in image classification.

In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyperplane that differentiate the two classes very well. The hyperplane must be that maximize minimum distance to the training examples. This distance receives the name of margin within SVM's theory. Therefore, the optimal separating hyperplane maximizes the margin of the training data. We can see a 2-dimensional example in Figure 2.7.

As we see in Figure 2.7, the simplest way to perform the separation is by a straight line. Unfortunately these cases are unusual for real applications and the SVM algorithm must deal with more than two variables, nonlinear separation curves, cases where data sets can not be completely separated or classifications in more than two categories. The representation through Kernel functions offers a solution to these problems, projecting the information to a feature space of greater dimension which increases the computational capacity of the linear learning machines. As shown in Figure 2.8, we will map the input space to a new feature space of greater dimensionality with the Kernel function, ϕ.

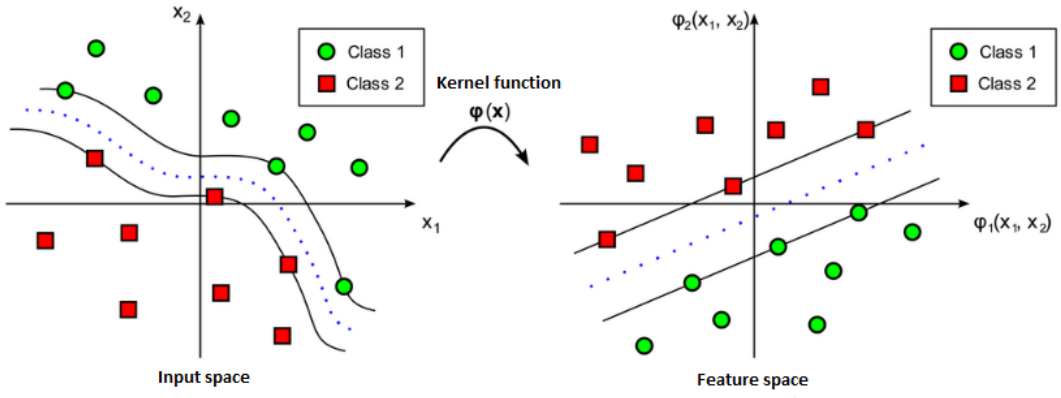


Figure 2.8: Example SVM Kernel function.

Note that it is not always possible to make a perfect separation and, if it is, the result of the model can not be generalized to other data. This is known as overfitting. In addition, for the training of the support vector machine to be correct, it is necessary that the samples of the classification categories to be balanced. This means that there must be the same number of samples of all categories, but we would get an unbalanced classifier.

2.5 Conclusions

In this chapter the main techniques used in the state of the art are exposed to achieve the goal of the project: classify a new video entry as violent or non-violent. As one can intuit, the developed algorithm will be structured as follows: detection and description of points of interest, extraction of the bag of words using kmeans, calculation of the individual histograms that will describe each video and, finally, training of the Support Vector Machine that will act as classifier. The following chapters will explain in detail the developed algorithm as well as the results obtained and the conclusions that can be established.

Chapter 3

Design and development

3.1 Introduction

In this chapter we will describe step by step the developed algorithm whose ultimate goal is classify a new video entry as violent or non-violent. For this, it will be necessary to train a classifier -support vector machine (SVM)- that, having the description of the new video, is able to label it correctly.

The developed algorithm follows the structure of Figure 3.1. First, it will be necessary to describe the appearance of each video frame through extraction and description of interest points. In parallel form, the motion will be estimated and the direction Lagrangian measures will be calculated and described for each point of interest detected. With this, we will have a complete description of the video sequences. Once this is done, we will calculate the vocabulary of Bag-of-Words model using k-means algorithm. This vocabulary will be used to encode each video by obtaining a fixed-length frequency histogram. With these histograms we will train the final classifier that will be able to label an entry video as violent or non-violent. All of this will be explained in detail in the following sections.

3.2 Appearance description

In computer vision, usually the appearance description of an image is carried out in two differentiated stages: extraction and description of interest points. In this section the algorithms proposed in each of these two stages will be described in detail. The algorithm developed uses the standard SIFT method for the interest points detection and, later, these key points are described by the SP-SIFT (Super-Pixel-based isolation of SIFT).

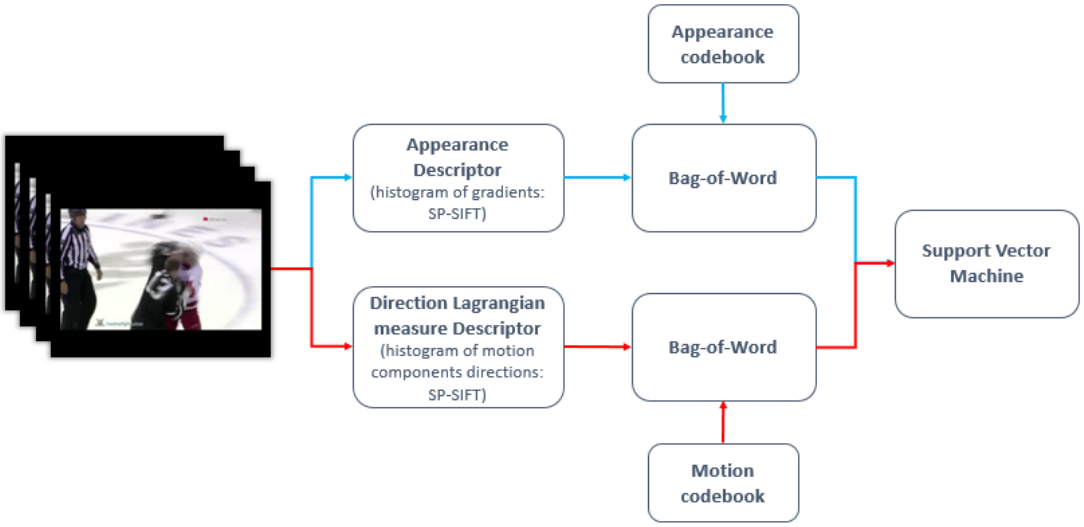


Figure 3.1: *Blocks diagram.*

3.2.1 Interest points extraction

As already mentioned in Section 2.2, the interest points detection consists in the location of points that, thanks to their characteristics, are able to maintain their shape despite changes in the image and, at the same time, contain relevant information about the image. That is, find those points that are invariant to typical transformations such as translations, rotations, changes of scale, perspective or illumination.

In this project the SIFT detector described by Lowe in [14] has been used. As explained in Subsection 2.2.1 of the State of Art, the SIFT detector uses the concept of Gaussian pyramids to obtain the differences between adjacent smoothed images (DoG), this can be seen in Figure 2.1a. Figure 2.1b shows the way of extracting interest point candidates by locating repeatable local maxima and minima at different scales of this space. For that, each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below. Finally, as explained in detail in Subsection 2.2.1, the original SIFT algorithm eliminates those points with poor stability and returns the real key points as well as the scale at which they have been detected.

In the Figure 3.2 we can see an example of the SIFT points detected in an image. Smaller circles means that the scale at which the point has been detected is small and greater circles are greater scales. The line within the circle indicates the main

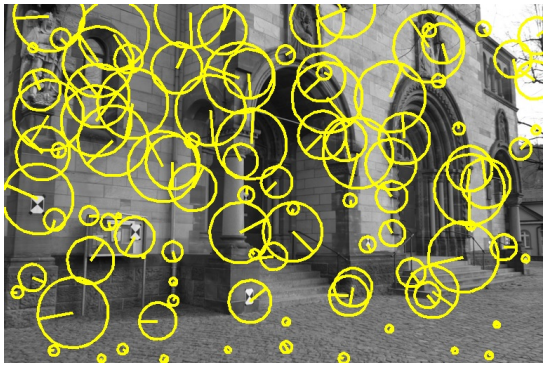


Figure 3.2: Example of SIFT points detected.

orientation of the key point.

The implementation of the interest points detection stage has been developed so that the user can introduce a matrix of personalized points for the algorithm to describe them, that is, there is no need for SIFT detection. This is useful for calculating the performance of any other detection methods for the violent video classification task.

3.2.2 Interest points description

Once the interest points are located, it is necessary to describe them. As we have already advanced, to describe the interest points has been used the SP-SIFT algorithm developed in the VPULab of the Universidad Autónoma de Madrid [21]. Since the SIFT points detected are usually found in the edges of image objects, the main advantage of using the SP-SIFT description is his ability to abstract the object information of the background in which it is found. This can play a key role, for example, in object recognition tasks.

The SP-SIFT description algorithm comes from a combination of the SIFT method with the segmentation-based methods known as Superpixels. This combination allows isolating in different descriptions information pertaining to different regions of the key point neighborhood. To achieve this the algorithm works as follows:

- **Splitting the image into Superpixels.** It is a complete division of the image, without exclusions and without overlap between regions. Regions known as Superpixels (SP) allow the generation of pixel clusters large enough to take on semantic meaning as compared to a single pixel, but small enough to achieve a good fit to the edges. As we can see in the Figure 3.3, Superpixels obtain

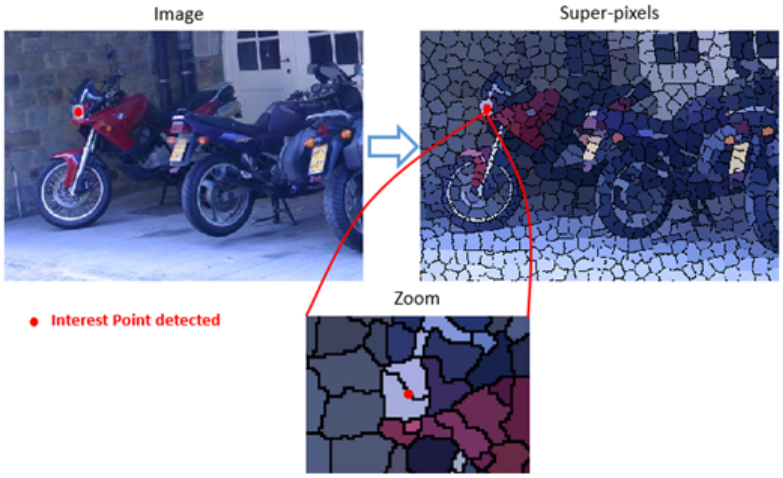


Figure 3.3: Superpixels segmentation example.

homogeneous regions on the image that are meaningless when describing or recognizing objects, instead, when we use this division together with the information of the original image and the SIFT descriptor we are able to describe each point of interest according to the information of the individual regions of its neighborhood. The SP-SIFT algorithm uses the implementation of Superpixels developed by Vedaldi [22].

- **Descriptor mask.** By using the regions extracted in the previous step, the original SP-SIFT description of the points of interest will be modified. In the original SP-SIFT algorithm, the interest point will be replicated as many times as the number of regions of equal or greater size to a threshold, that are found in the neighborhood (see Figure 3.4). In our case, and in order to eliminate any information about the background of the image, only the pixels of the neighborhood that are within the region where the point of interest is located will be considered, canceling the rest of the pixels in the neighborhood. An example of this can be seen in Figure 3.5, where the interest point is located in the nose of the dog and, after split the image into superpixels, it is possible to describe only the complete nose making zero the rest of the neighborhood gradients of the key point.

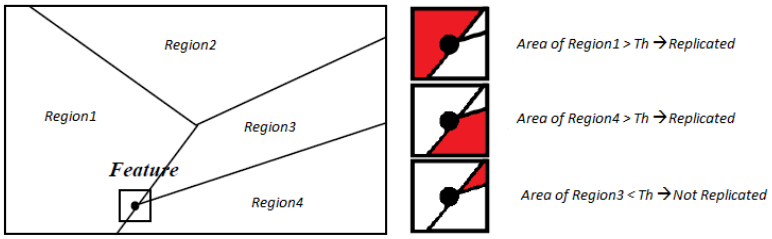
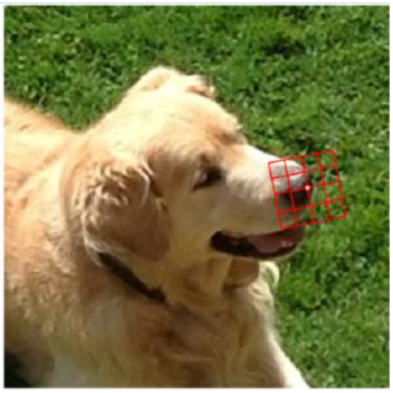


Figure 3.4: Replication stage of the original SP-SIFT algorithm.



(a) Original image.

(b) Description region.

Figure 3.5: Example SP-SIFT description.

- **Histogram calculation.** The histogram calculation corresponds with the generation of the descriptor. At this stage, it is looking to cancel in the histogram the influence of the information existing outside the previously extracted region through the use of Superpixels. To do this, and in order to maintain the invariance to rotations and linear translations that has the original SIFT algorithm, the main orientation is calculated taking into account only the region of the key point, that is, canceling the information of the pixels that are not in this region. Once this has been done, the descriptor is calculated as seen in Subsection 2.2.1 of the State of Art.

3.3 Motion description

The estimation of motion between frames has become fundamental as far as video analysis is concerned. More specifically, in the classification of violent events the

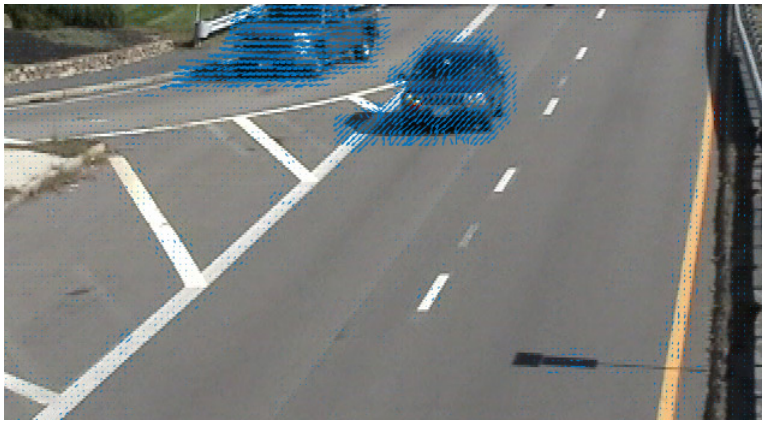


Figure 3.6: Optical flow example.

information provided by the movement is essential in order to make a correct classification. There are numerous algorithms based on the extraction of the optical flow fields, in the next sections the concept of optimal flow as well as the direction Lagrangian measures are used in our algorithm and the way of describing them will be explained.

3.3.1 Optical flow field

Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene. The optical flow methods try to calculate the motion between two consecutive frames. Optical flow is used to compute the motion of the pixels of an image sequence. It provides a dense (point to point) pixel correspondence. The optical flow is formed by the components x and y of the motion of the image. It would be represented as shown in the example in Figure 3.6, showing the movement of the pixel in the initial frame to its position in the final frame. As already mentioned in the State of Art chapter, most algorithms exist in the violent sequences classification, use this type of movement measures to describe their videos taking into account only two frames.

It is essential that the optical flow algorithm overcomes the many challenges that arise in realistic videos, namely: robustness to outliers (motion discontinuities, occlusions), robustness to illumination changes (with gradient constancy), ability to deal with large displacements.

At this stage of the project we have used an executable provided by Tobias Sents and the Technical University of Berlin where we have the option to execute various

algorithms that calculate the optical flow of the videos. In our case, the method chosen is Deep Flow [23] which, despite being slower, has a higher performance according to the classification of a Database and Evaluation Methodology for Optical Flow, published open access in International Journal of Computer Vision, [24]. Other methods of calculation of optical flow are DualTVL1 or disflow.

The executable file used returns an .avi file where the motion of the video can be viewed and a treatable .ofl file where the motion information is stored. A Matlab function has been created able to read this type of files and extract the x and y components of each pixel during the entire video sequence. The computational time spent reading this file is high and will be identical for any of the chosen optical flow calculation methods (deepflow, DualTVL1, disflow ...). In addition, the calculation of the optical flow will only be done once on the chosen dataset. For these two reasons, it has been decided to use the deepflow method because, although it is slower, it has a better performance.

3.3.2 The direction Lagrangian measures extraction

As discussed in the previous subsection, the optical flow fields only take into account two frames of the video sequence to estimate the motion. In this way, only short events can be considered. Algorithms named in Chapter 2 (MoSIFT, STIP or ViF) use only this type of motion information, losing visibility in longer duration events. That is to say, an event that is composed of long term motion information will not be encoded completely but by its short time single parts.

The algorithm developed intends to take into account more than two frames to extract the appropriate motion information. For this, we have used the Lagrangian theory approach described in [18, 25], which takes into account as many frames as the integration parameter, τ.

The Lagrangian theory provides a set of tools for the analysis of continuous motion based on integral field lines similar to trajectories. This theory has its origin in the analysis of general dynamical systems and are fundamental in computational fluid dynamics systems. Lagrangian analysis is based on particle trajectories in the space-time domain. In this context, the concept of Lagrangian Coherent Structures (LCS) is a powerful tool to describe temporally complex spatial motion hidden patterns.

Within the video analysis these methods characterize the motion dynamics and achieve a time dependent vector field with the help of the optical flow fields. Therefore, the performance of this tools will depend on the optical flow estimation. The flow map, $\phi(x, t_0, \tau)$, maps an initial point x at time t_0 to its positions after the time τ. Usually, this time is only two frames as we can see in Figure 3.7.

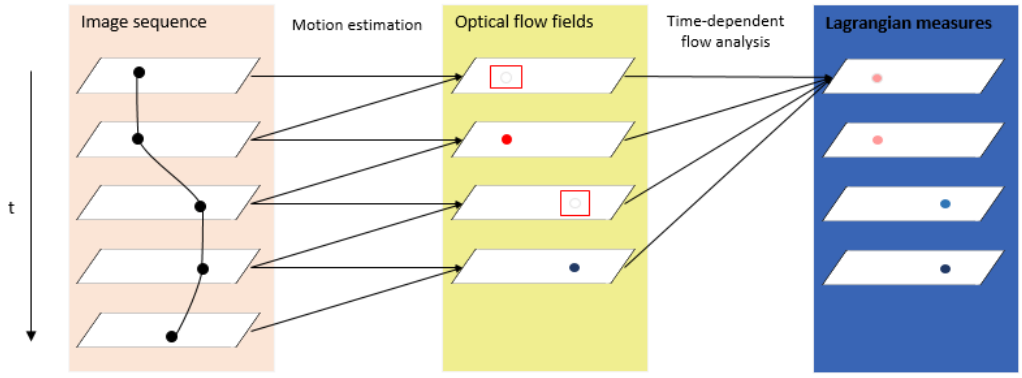


Figure 3.7: Concept of Lagrangian measures.

Following the Lagrangian framework proposed in [25] we use the path lines concept to get the trajectories of each pixel of the image between two instants of time. Path lines are polynomial that obtained by combining all position in the flow map for a specific point over a time interval, $[t_0, t_0 + \tau]$. In mathematical terms, we can described path line for a space-time point (x_0, t_0) as follows:

$$\frac{d}{dt}\begin{pmatrix} x \\ t \end{pmatrix} = \begin{pmatrix} v\left(x\left(t\right), t\right) \\ 1 \end{pmatrix}, \begin{pmatrix} x \\ t \end{pmatrix}(0) = \begin{pmatrix} x_0 \\ t_0 \end{pmatrix}$$

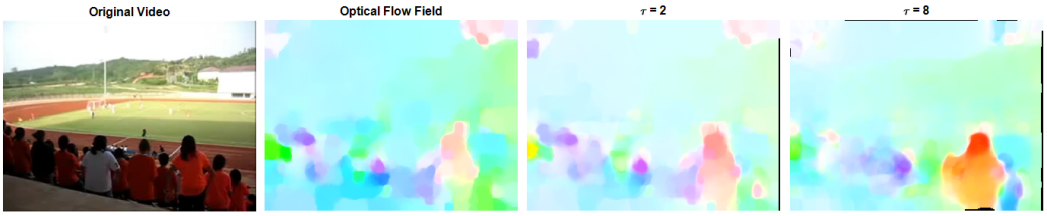
where $v = (x, t)$ is the optical flow vector field.

As explained in [25] exists different types of Lagrangian measures such as arc length, separation or direction. In this work we use the direction Lagrangian measure that can be obtained by accumulating the average motion vector along the path lines. We can formulate it as:

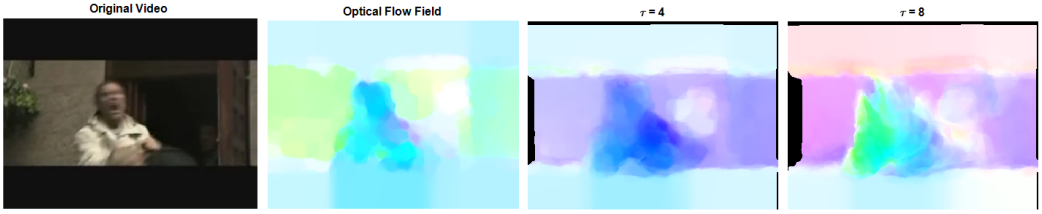
$$\Lambda_{x/y}\left(x, t_0\right) = \frac{1}{\tau}\int v_{x/y}\left(\phi\left(x, t_0, \tau\right)\right)\partial\tau$$

Where $v_{x/y}$ is a function providing the x or y motion components. A key aspect is the choice of the time interval parameter, τ, that defines how many frames are considered for the Lagrangian feature. This parameter allows to describe the movement in different temporal scales. The functionality of this parameter can be seen in Figure 3.8[1], where the integration parameter allows us to describe better a fast movement, such as boxing, with a low τ (3.8b). And slower actions, such as dance, will be better described with a high τ (3.8a).

[1] Annex 1 explains the color code used to represent the movement in the images.



(a) High τ to describe slow actions.



(b) Low τ to describe quick actions.

Figure 3.8: Example of the integration parameter influence, τ.

Therefore, focusing on the development of our algorithm, to estimate the direction Lagrangian measures we started from the optimal flow fields extracted in the previous step using Deep Flow. These fields do not usually contain integers, ie, the displacement of a pixel from its initial position to its position after a frame does not correspond to an exact matrix cell of the Matlab image. Because of this, it has been necessary to perform bilinear interpolations to calculate the trajectories corresponding to each pixel between several frames. The purpose of the interpolations is to extract the value of the pixel after the motion given by the optical flow to obtain the path and to continue analyzing it for several frames.

3.3.3 Global motion compensation

As we have already advanced several times, the reliability of the Lagrangian measures are mainly based on the quality of the calculated optical flow. Camera motion has a significant impact to the motion signature of the directional field, therefore, in order to try to improve these measures as far as possible, a global motion compensation has been made.

To do that we extract a homography matrix, H, using a grid of points over the image (the number of points used is configurable by the user), $\begin{bmatrix} x_{grid} & y_{grid} \end{bmatrix}$, and looking for their correspondences through the estimated direction Lagrangian measures obtaining the displacement of each one of the points of the grid (see Figure

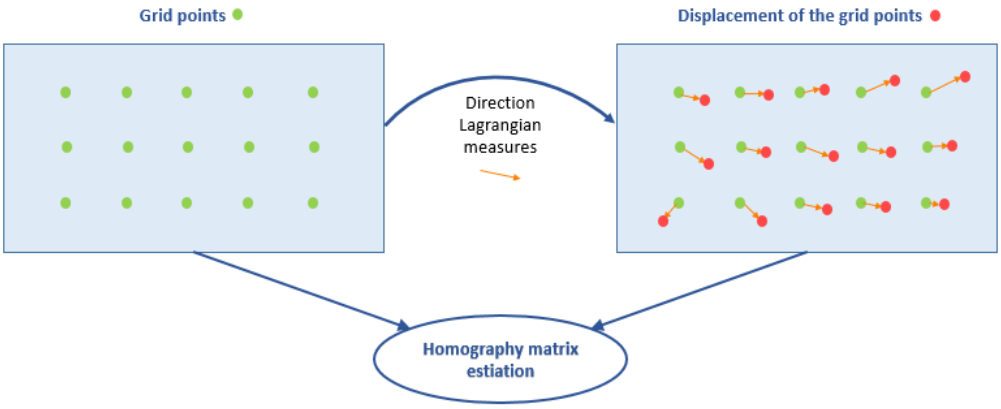


Figure 3.9: Homography matrix estimation.

3.9). We calculate the homography matrix through the points of the grid and their correspondences. The higher the number of points used the better the homography estimation will be, but the algorithm will be slower.

The global motion, $\begin{bmatrix} x_{motion} & y_{motion} \end{bmatrix}$, is obtained by multiplying the homography matrix by the current directional field:

$$\begin{bmatrix} x_{motion} \\ y_{motion} \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} x_{grid} \\ y_{grid} \\ 1 \end{bmatrix} \tag{3.1}$$

Finally the compensated direction field can be found by subtracting the global motion from the actual field. We can see an example of this development in Figure 3.10.

3.3.4 The direction Lagrangian measures description

Previously, in Subsection 3.2.2, the operation of the SP-SIFT descriptor was explained. In this case, in order to describe the Lagrangian measures we will use the same descriptor, adapting it if necessary.

Lagrangian measures are composed by the motion component of the pixel on the x-axis and the motion component on the y-axis. Given this configuration, the Lagrangian measures can be treated in the same way as the gradients that are extracted

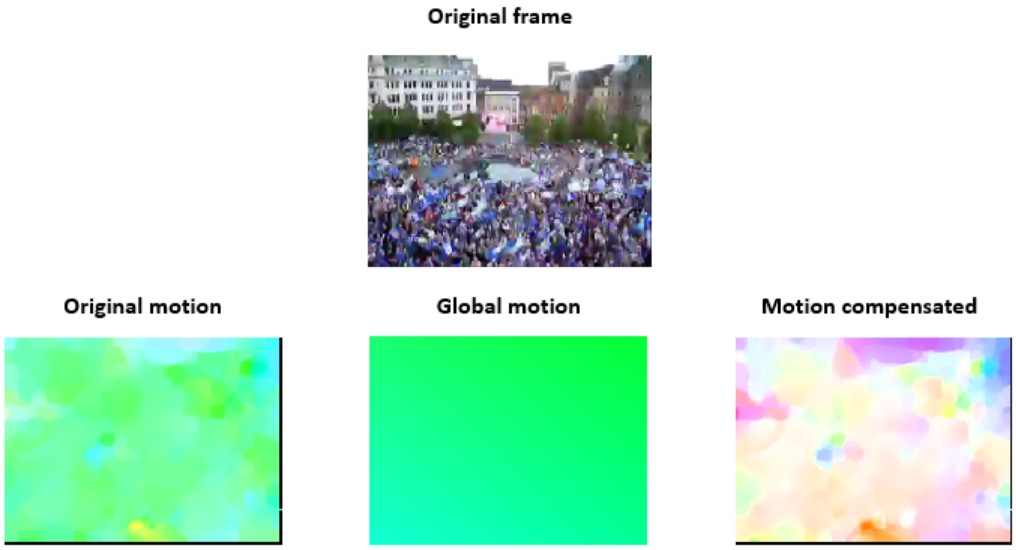


Figure 3.10: Motion compensation example

to calculate the SIFT descriptor. So the extraction of the SP-SIFT descriptor for the Lagrangian measures will take the following steps:

- **Splitting the image into Superpixels.** At this stage the original frame is completely divided into regions, without exclusions and without overlap between regions in the same way as for the calculation of the original SP-SIFT descriptor.

- **Descriptor mask.** Again, this stage is implemented as for the SP-SIFT appearance descriptor (see Subsection 3.2.2).

- **Histogram calculation.** As we have seen, the histogram calculation step corresponds with the generation of the descriptor. As in the SP-SIFT descriptor of appearance, it is looking to cancel in the histogram the influence of the information existing outside the previously extracted region through the use of Superpixels. In Section 2.2 of the State of the Art, it was explained how the SIFT descriptor use the gradient for the calculation of the descriptor. This was used for the extraction of the appearance descriptors. In this case, to describe the Lagrangian measures will not be taken into account the gradients but the motion components. The SIFT algorithm performs all the original steps but with the motion information instead of the gradient information.

3.4 Bag-of-Words model

Currently this framework is widely used for object recognition based on features extraction. As already explained in the Section 2.3 of the Chapter of the State of the Art, the idea is to extract a set of vectors of representative characteristics by means of the use of a clustering algorithm and, later, encode each video using that set of vectors called dictionary or vocabulary of the word bag model.

At this stage of the project we have used an executable provided by Tobias Sents and the Technical University of Berlin. The k-means algorithm was used to calculate the representative vector of the Bag-of-Words. It is possible to modify parameters in this executable file, for example the number of clusters to be extracted or the percentage of input data that will be used for its calculation. In our case, the number of clusters used has been 800.

In order to carry out a posterior cross validation and to avoid overfitting of the support vector machine, five different dictionaries are extracted through this process. To do this the complete dataset is divided into five folders containing the same number of violent and non-violent videos. Four of these five folders will be used as input data to calculate possible dictionary. That is, for the first dictionary in the first folder will be left out of the calculation, for the second dictionary in the second folder will not be taken into account, and so on.

Once the five dictionaries are obtained, we will extract five frequency vectors for each video (each calculated with a different dictionary). The calculation of the frequency vectors is based on taking each descriptor of the video, calculating the Euclidean distance with each of the representative vectors of the dictionary, and assigning the one whose distance is smaller, that is, the one that best represents it.

With this technique we can reduce the dimensionality of the problem. In the previous stage, for each video there were many feature descriptors, but now, each video can be represented with a single fixed-lenght vector. This fixed length will be equal to the number of clusters chosen when running the k-means clustering.

3.5 Support Vector Machine

As explained in Section 2.4 of the State of the Art, support vector machines are a set of supervised learning algorithms normally related to classification and regression problems. As we have seen, the basic idea of an SVM is to construct a hyperplane or set of hyperplanes in a space of several dimensions that can be used to separate, in an optimal and unequivocal manner, each of the input classes. These hyperplanes

will define our SVM in such a way that for an unknown input data it is able to assign it the category to which it belongs. While most of the learning methods focus on minimizing the errors made by the model generated from the training data, this type of algorithm looks for the separation hyperplane that equidistant from the closest training samples of each class. This is called the maximum margin and therefore the SVMs are sometimes known as maximum margin classifiers. In this way, the points that are labeled with one category will be on one side of the hyperplane and the cases that are in the other category will be on the other side, separating both classes clearly. The ideal case, as already mentioned in the corresponding section of the state of art, is that in which we have only two classes and these can be separated by a single plane. Usually, real application cases are not so simple because there are more than two classes, the separation curves are not linear or directly the data can not be separated completely. Due to these limitations, the representation through kernel functions arises. These functions map the information to what is known as the characteristic space. This feature space is usually of greater dimensionality but simpler for complete separation of data (see Section 2.4).

By focusing on the development of our project, to proceed to the training of this type of machine, in the first place, it is necessary to have a set of training data correctly labeled. In our case, the two possible classes or tags are video "Violent" or "Nonviolent". Recapping, thanks to the previous stage we have a single fixed-length vector that serves as a description of a complete video. In turn, the video is labeled as violent or non-violent and both the fixed-length vector and the tag will be input to the training function of the Support Vector Machine.

Once the Support Vector Machine has been trained, we can check its performance through the description of unknown videos. The form of evaluation that has been used will be seen in Chapter 4.

Chapter 4

Test and results

In this chapter we will introduce both the dataset and the code used to achieve the main goal of this project. In addition, we will explain the evaluation process that has been carried out and analyze and compare our results with the other algorithms results like MoSIFT or LaSIFT explained in previous chapters. We will also analyze the influence of the camera global motion compensation and the integration parameter -τ- in the direction Lagrangian measures.

4.1 Datasets and code

In order to evaluate the performance of our algorithm we have focused on the use of a single dataset. This dataset has already been used previously to evaluate algorithms that pursue the same objective as we, such as ViF or LaSIFT.

Violent Crowd dataset created by Hassner *et al.* [8]. This dataset contain 246 short sequences with different crowded scenarios such as bars, or football stadiums. The videos, extracted from Youtube, have a 320x240 resolution and are labeled manually as violent or not violent. The quality, the movement of the camera, the different scenarios, the change of lighting or the crowd make this dataset a challenge for the detection of violence.

In addition, and as we have already advanced in Chapter 3, for the development of this work several sources of code have been used. Among these sources we can highlight the original SP-SIFT code developed at the Universidad Autónoma de Madrid. Within this code, we work with the `http://www.vlfeat.org/` toolbox that has been used to extract SIFT key points and construct the SP-SIFT descriptor. In addition, the calculation of this descriptor requires the use of the Superpixels implementation of [22]. It should be noted that the default parameters of the SP-SIFT

descriptor have been maintained, including the size of the Superpixels or the relative importance of the border versus the color terms.

Technische Universität Berlin has provided us code repositories for both the estimation of optical flow and for obtaining dictionaries from the word bag model.

To carry out the global motion compensation of the camera it is necessary to calculate the homographic matrix between the correspondences produced between a grid and that same grid added to the apparent movement given by the Lagrnagian measurements. To calculate this homography, the code provided by the authors of [26] in `https://www.robots.ox.ac.uk/~vgg/hzbook/` has been used.

Finally, to train and test the Support Vector Machine, the Matlab code of the LIBSVM repository has been used.

As we see, most of the algorithm is implemented in Matlab code.

4.2 Evaluation procedure

To evaluate the results of the algorithm developed, the cross-validation method was used. This technique is based on repeating the experiment several times with different partitions of the training set and test and finally calculating the arithmetic mean of these experiments (see Figure 4.1).

This type of validation is used in environments where the main objective is the prediction and we want to estimate the accuracy of the model generated. The main advantage of this technique is to ensure that the results of the model are independent of the partition between training data and test data.

Obtaining a classification model through a set of training data consists of uniquely separating that set into as many classes as is established. This division may lead to overfitting to training data. Therefore, in order to achieve reliable results, different classifier models with different training data are obtained. These models will be evaluated with the remaining data set, called test data.

In Figure 4.1 we can see the actual process carried out in the evaluation of our algorithm. The total data set has been divided into 5 folders. Each of the folders will have the same number of data labeled as violent and non-violent. In this way, the training of the classifier or support vector machine will be balanced, avoiding the performance delay to one of the classes. In addition, each individual classifier model will use a dictionary calculated only with the training data to be used. Thus, the influence of the test data will be null throughout the process and the final results will be more reliable since the test videos will be totally unknown to the classifier.

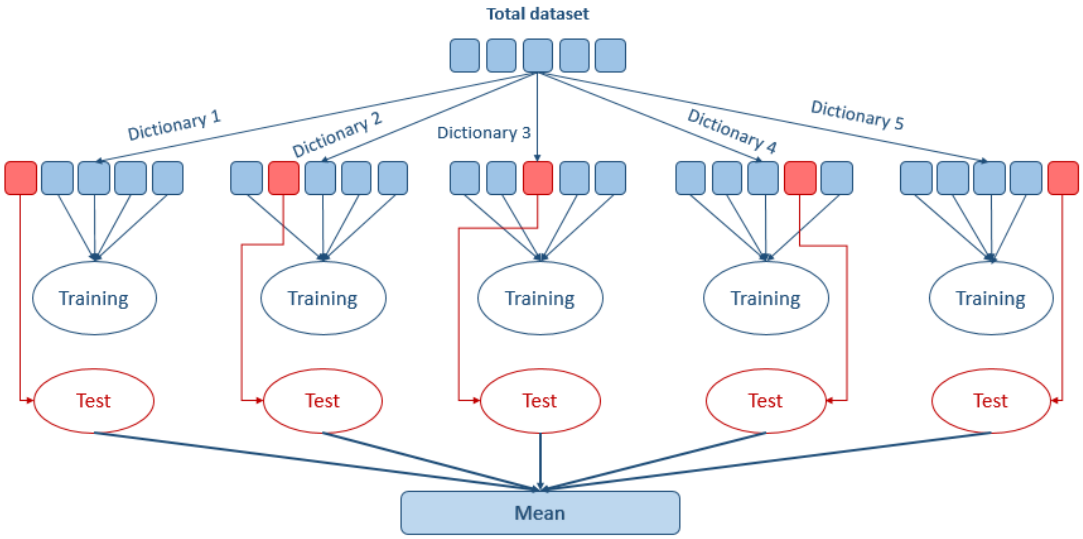


Figure 4.1: Cross-validation.

4.3 Measures of error

As we have already advanced in Section 4.2, the main measure of error that we are going to use is given by the arithmetic mean existing after the cross-validation. This result will be given as the mean of the percentage of success among the 5 trained classifier models. It has been decided to use this measure of error as a basis for the calculation of the performance because it is used in reference works for the development of our algorithm like, for example [18].

In order to obtain more detail on the origin of the errors made, the confusion matrix will be extracted from each experiment and the ROC curve will be calculated.

A Receiver Operating Characteristic (ROC) curve is the representation of the True Positive Ratio (TPR) versus the False Positive Ratio (FPR). This representation is used to evaluate the performance of binary classifiers such as ours in which the output for a new input video can only be "Violent" or "Non-violent". To understand this representation we have to classify the results into what is known as the confusion matrix. As we can see in the example in Figure 4.2, the confusion matrix divides the results into False Positives, False Negatives, True Positives and True Negatives.

Let's look at an example. Consider a binary class prediction problem, like ours. The results are labeled positive (P or, in our case, Violent) or negative (N or non-violent). There will be four possible results from this classifier. If the result of a prediction is positive and the actual value is also positive, then it is known as a True Positive (TP); However if the actual value is negative then it is known as a False

Positive (FP). Likewise, we will have a True Negative (TN) when both the prediction and the real value are negative, and a False Negative (FN) when the result of the prediction is negative but the actual value is positive.

A False Positive will occur when interclass variability is low. That is, samples of different classes obtain very similar results, so they are difficult to differentiate. On the other hand, a False Negative occurs when intraclass variability is high. This means that two samples of the same class have very distant results between them.

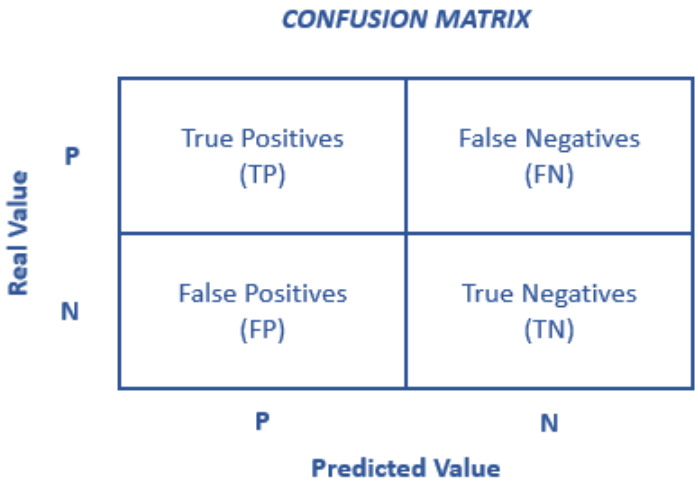


Figure 4.2: Confusion Matrix

To draw a ROC curve only the True Positive (TPR) and False Positive (FPR) ratios are necessary. The TPR or True Positives Ratio measures the extent to which a classifier is able to classify positive cases correctly, out of all the positive cases available during the test. The FPR or False Positives Ratio defines how many positive results are incorrect among all negative cases available during the test. In Figure 4.3, we can see what a ROC curve would look like, on the x-axis we would represent the False Positive Ratio and in the axis and the True Positive Ratio or also know as sensitivity.

Analyzing this graph, the best possible method of prediction or perfect classification point would be placed in a point in the upper left corner - coordinated (0,1) - representing a 100% sensitivity, that is, no false negatives, and a 100% also of specificity, no false positives. On the other hand, a totally random classifier would be represented by a point on the diagonal that crosses the graph. In this case we can find examples of experiments like throwing a coin. Finally, the points that remain

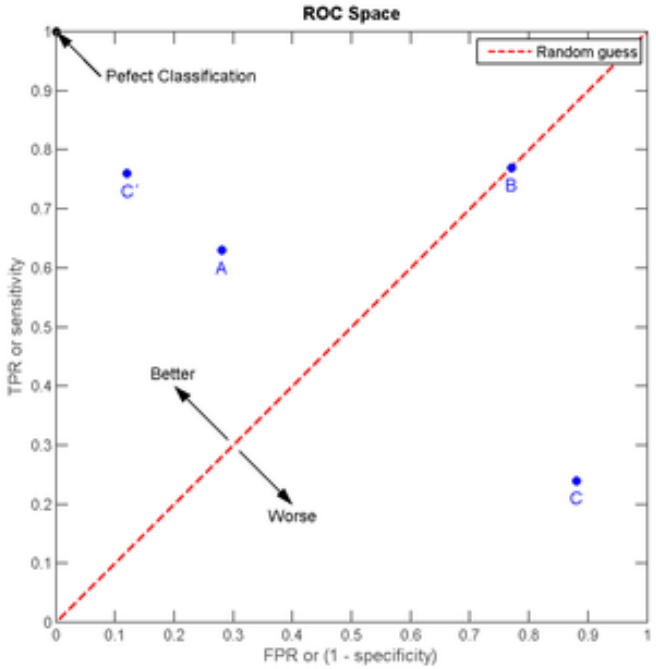


Figure 4.3: ROC curve.

above the diagonal would represent the performance of better classifiers than chance and below the diagonal would be the classifiers worse than chance. In case the results of the classifier were below the diagonal of the graph, it would be enough to reverse the result of the classification in such a way that the positive cases became negative and vice versa to improve the performance.

Explained this, we have the basic concepts needed to understand the analysis of the results shown below.

4.4 Results analysis

Throughout this Section the results obtained through different experiments will be shown and analyzed in a comparative way. For this purpose, the evaluation form explained above and the error measures set out in the previous Section will be used.

To show the results orderly, we will start by looking at the tests performed with SIFT and SP-SIFT as methods of describing the appearance of the video. In this case,

			SVM kernel	
Appearance descriptor	Motion descriptor	¿Global motion compensation?	Chi-Squared	Intersection
SIFT	--	--	88,68%	87,87%
SP-SIFT	--	--	89,88%	86,23%

Table 4.1: Appearance description: SIFT vs SP-SIFT

the motion will not be estimated yet, so the only information that the classifier will have to decide between the labels "Violent" or "Non-violent" will be the appearance of each of the frames.

Subsequently, the description, by means of SIFT and SP-SIFT, of the movement calculated with the Deep Flow method will be evaluated in a comparative way. In this case, both the previous description of appearance and the new description of the estimated motion are included. Here you will see the results obtained with MoSIFT which, as we saw in Subsection 2.2.2, uses the SIFT descriptor to extract both the characteristic vector of the appearance of the frames and that of the motion.

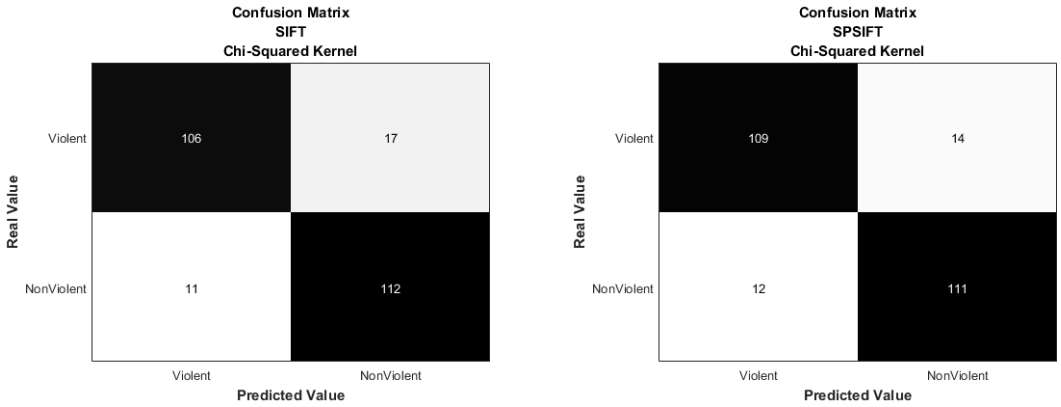
We will continue with the bulk of the results of our algorithm. For that, we use SP-SIFT for the description of the appearance and we will calculate the Lagrangian measures with different integration parameters τ. This measures will be described with both SIFT and SP-SIFT. With this, we will be able to draw conclusions about the use of the integration parameter and we will also evaluate the use of the direction Lagrangian measures against the classic use of optical flow fields.

Finally, we will include a last evaluation including the phase of camera global motion compensation.

4.4.1 Appearance description: SIFT vs SP-SIFT

As we have already commented in the introduction of this Section, firstly, we will not estimate the motion of the video since we only want to compare the performance of the descriptors of appearance. The results are shown in Table 4.1 where, although the performance of both feature descriptors is similar, SP-SIFT tends to have a higher percentage success when the Support Vector Machine kernel is Chi-Squared and SIFT outperforms SP-SIFT when using Intersection kernel.

In addition, in Figure 4.4 we can see examples of confusion matrices of the previous



(a) SIFT, Chi-Squared Kernel (b) SP-SIFT, Chi-Squared Kernel

Figure 4.4: Confusion matrix of appearance descriptor evaluation.

experiments. [1]. In the case of SIFT (Figure 4.4a) we show the confusion matrix of the experiment carried out by calculating the classifier model with the Chi-Squared kernel. And, in 4.4b we can see the SP-SIFT confusion matrix whose classifier model has also been calculated with a Chi-Squared kernel.

With the confusion matrices we can see where the errors of both systems come from. On the one hand, in terms of hits, we see that the classifier using the SP-SIFT appearance descriptor is able to correctly label a greater number of violent videos (True Positives), 109 vs 106 SIFT. In contrast, SIFT has a better hit rate for correct detection of non-violent sequences (True Negatives), 112 vs 111 SP-SIFT. As for the class of errors committed by each of the algorithms, we have similar rates of False Positives in both methods (11 FP SIFT vs 12 FP SP-SIFT), as we have already seen, this means that the video has been classified as violent when in fact it is nonviolent. We find a greater difference in the False Negative rate where the SP-SIFT algorithm only commits 14 errors and SIFT three more.

At first sight, with these results we can not draw any conclusions since it depends on the type of kernel used to calculate the support vector machine.

4.4.2 Deep Flow description: SIFT vs SP-SIFT

In this Subsection we want to compare the influence of the estimation and description of the apparent movement of the scene with respect to the absence of this. For this,

[1] The rest of confusion matrices are found in the Annex B

			SVM kernel	
Appearance descriptor	Motion descriptor	¿Global motion compensation?	Chi-Squared	Intersection
SIFT	SIFT	x	91,93%	89,48%
SIFT	SP-SIFT	x	89,07%	90,28%
SP-SIFT	SIFT	x	90,65%	89,43%
SP-SIFT	SP-SIFT	x	89,47%	89,45%

Table 4.2: Deep Flow description: SIFT vs SP-SIFT

the motion between two consecutive frames has been calculated by means of Deep Flow and it has been described with both the SIFT algorithm and SP-SIFT to extract the best possible conclusions.

Using SIFT both to describe the appearance of the frames and the optical flow field would be equivalent to the procedure performed by MoSIFT (see Subsection 2.2.2).

First, and as we can see in Table 4.2, the best results are usually obtained using a Support Vector Machine with Chi-Squared kernel. Focusing on this column we see that the method that is equivalent to the algorithm MoSIFT - first row - is the one that has better rates of success against the other methods. In second position we would have the method that uses SP-SIFT as descriptor of appearance and SIFT as descriptor of movement. And the worst result is obtained when using SIFT to describe the appearance of the frames and SP-SIFT to characterize the movement.

Analyzing the two tables of results seen so far, 4.1 and 4.2, we can conclude that the SP-SIFT algorithm performs better than SIFT in terms of appearance description but when used for describing the motion of the scenes its performance is less than that of its predecessor algorithm (SIFT).

4.4.3 Direction Lagrangian measures description: SIFT vs SP-SIFT

In this case, the Lagrangian measures have been calculated with the aid of the previous optical flow (Deep Flow). As we saw in Subsection 3.3.2, the direction Lagrangian measures are characterized by the number of frames they take into account. This number of frames is represented by the integration parameter, τ. In the following tables we can see the evolution of the performance of the classifiers depending on the chosen parameter τ.

			SVM kernel	
Appearance descriptor	Motion descriptor	¿Global motion compensation?	Chi-Squared	Intersection
SP-SIFT	LaSIFT $\tau = 3$	x	90,28%	91,47%
SP-SIFT	LaSIFT $\tau = 4$	x	91,50%	90,28%
SP-SIFT	LaSIFT $\tau = 5$	x	91,93%	90,27%
SP-SIFT	LaSIFT $\tau = 6$	x	91,92%	89,05%
SP-SIFT	LaSIFT $\tau = 8$	x	91,10%	90,72%

Table 4.3: Direction Lagrangian measures without global motion compensation described by SIFT.

			SVM kernel	
Appearance descriptor	Motion descriptor	¿Global motion compensation?	Chi-Squared	Intersection
SP-SIFT	LaSP-SIFT $\tau = 3$	x	89,47%	89,47%
SP-SIFT	LaSP-SIFT $\tau = 4$	x	89,88%	88,25%
SP-SIFT	LaSP-SIFT $\tau = 5$	x	89,88%	87,82%
SP-SIFT	LaSP-SIFT $\tau = 6$	x	91,10%	90,25%
SP-SIFT	LaSP-SIFT $\tau = 8$	x	91,10%	89,07%

Table 4.4: Direction Lagrangian measures without global motion compensation described by SP-SIFT.

As we can see in Tables 4.3 and 4.4 all the experiments describe their appearance with SP-SIFT. On the one hand, Table 4.3 use SIFT to describe the direction Lagrangian measures and, on the other hand, the table 4.4 uses the SP-SIFT method.

As we can see, and regardless of the kernel used in the training of the SVM, the performance of the SP-SIFT descriptor is inferior to that of the SIFT method in the description of the measures of Lagrangian.

If we analyze the influence of the integration parameter, we see that the best results are found around the values 5 and 6. This means that the most characteristic actions in order to recognize violent sequences are between 5-6 frames. If only short events are taken into account as happened with optical flow, which only analyzed two frames in the movement, these actions characteristic of the violent sequences are not

			SVM kernel	
Appearance descriptor	Motion descriptor	¿Global motion compensation?	Chi-Squared	Intersection
SP-SIFT	LaSIFT $\tau = 3$	✓	89,88%	89,08%
SP-SIFT	LaSIFT $\tau = 4$	✓	90,54%	90,95%
SP-SIFT	LaSIFT $\tau = 5$	✓	90,98%	89,29%
SP-SIFT	LaSIFT $\tau = 6$	✓	89,88%	89,88%
SP-SIFT	LaSIFT $\tau = 8$	✓	90,25%	88,60%

Table 4.5: Direction Lagrangian measures with global motion compensation described by SIFT.

covered, and therefore the performance of the classifier is smaller as can be observed comparing these two Tables with the Table 4.2. This conclusion about the optimum integration parameter to classify violent sequences was extracted by Tobias Senst *et al.* [18].

Finally, referring to the kernel used in the Vector Support Machine, in all the tests seen so far the Chi-Squared Kernel presents better results.

4.4.4 Influence of global motion compensation

In this Subsection we analyze the same experiments as in the previous one but using a compensation phase of the global camera movement. Again, tests were performed with different values of the integration parameter, τ.

As we can see in the Tables 4.5 and 4.6, using SIFT to describe the direction Lagrangian measures after the motion compensation the best results are obtained with $\tau = 5$. On the other hand, using SP-SIFT, the best performance is achieved with $\tau = 6$ or $\tau = 8$. This agrees with the conclusions drawn from the previous subsection.

In addition, if we compare these results with those obtained without motion compensation (Tables 4.3 and 4.4) we can see that in most cases the performance of previous systems can not be improved. This can be due to the fact that the dataset used presents many challenges that can affect the motion compensation phase implemented, such as environments with large crowds, low quality, etc ...

This phase significantly increases the execution time of the algorithm so it would not be optimal to include it as part of the final algorithm.

			SVM kernel	
Appearance descriptor	Motion descriptor	¿Global motion compensation?	Chi-Squared	Intersection
SP-SIFT	LaSP-SIFT $\tau = 3$	✓	88,63%	89,42%
SP-SIFT	LaSP-SIFT $\tau = 4$	✓	90,27%	88,98%
SP-SIFT	LaSP-SIFT $\tau = 5$	✓	89,05%	90,65%
SP-SIFT	LaSP-SIFT $\tau = 6$	✓	91,10%	88,25%
SP-SIFT	LaSP-SIFT $\tau = 8$	✓	91,10%	87,80%

Table 4.6: Direction Lagrangian measures with global motion compensation described by SP-SIFT.

4.5 Evaluation of results

In view of the previous tables we can evaluate both the operation of the SIFT and SPSIFT descriptors in the tasks of description of the direction Lagrnagian measures, the frames appearance or the characterization of the movement.

In addition, we can draw conclusions about the use of the different methods tested as well as the use of the implementation of the compensation of apparent camera motion.

Finally, in order to draw conclusions about the use of the Lagrangia measures for the concrete task of recognizing violent sequences, we evaluate the use of the integration parameter, τ, and its influence on the success rate of the methods that use these measures to estimate the movement of scenes.

The conclusions of the named aspects are shown below:

- Apparently, better descriptions of the appearance of the frames are obtained by the SP-SIFT descirptor (see Table 4.1). This conclusion is drawn after verifying that the best results are given by the calculation of the Vector Support Machine using the Chi-Squared kernel. In [21] it was already demonstrated that the use of SP-SIFT could increase the performance of SIFT in tasks such as object recognition in static images.

- When we have information about the apparent motion of the scene, the performance of the classifiers increases independently of the descriptor used to describe both the appearance of the video frames and the motion (see Table 4.2). It should be noted that, in this case, the SIFT descriptor is capable of obtaining

better results. This may be because the background information - discarded by SP-SIFT - helps to characterize the movement of the scene and, therefore, helps to discriminate between violence and non-violence.

- When we introduce the direction Lagrangian measures we consider more than two frames in the motion estimate. In the previous Tables we see how this consideration increases the performance of the classifier until reaching its maximum around τ equal to 5-6. This can be interpreted as that the detection of violence is an action that requires about 5-6 frames to be correctly detected. That is, it is not a slow action that needs more than 10 frames but it is not an action fast enough to be detected only with the use of 2 frames by calculating the optical flow fields.

- Finally, as for the influence of the global motion compensation of the camera, the results have not been as expected, since in most cases they are worse than those obtained without global motion compensation. In addition to obtaining worse results, the computational time required for the calculation of the homography matrices that compensate the movement significantly increase the total execution time of the algorithm. These poor results may be due to the difficulties presented by the dataset used.

Chapter 5

Conclusions and future work

5.1 Conclusions

This work aimed to obtain a classifier capable of discriminating between violent and non-violent videos using only the information of the frames of the video sequences, without additional information such as audio, subtitles or any other context information. In order to achieve this, different tools and algorithms of computer vision have been used, whose performance has increased in recent years thanks to the automation of processes like the one that is treated in this project.

To achieve this goal, a detailed study of the state of the art has been carried out. Within this study we have reviewed methods that pursue the same objective as ours, such as ViF, MoSIFT or LaSIFT. There has also been a previous study of the methods of detection and description of interest points as well as the basic concepts of motion extraction between frames of a video and the algorithms existing for that extraction.

After this study, and due to the good results obtained in its publication, it was decided to use the algorithm SP-SIFT developed in the VPULab of the Universidad Autónoma de Madrid. In this way we could measure the performance against its predecessor algorithm, SIFT, in a different task than the recognition of objects in static images. In the same way, thanks to the collaboration of the Technische Universität Berlin, it was decided to use the Lagrangian theory in estimating the motion between the frames of each video.

In summary, the algorithm developed can be divided into three clearly differentiated phases. In the first one, the interest points of each frame will be extracted and described, and the movement of the scene will be estimated and described by the use of the direction Lagrangian measures. In order to measure the performance in a com-

parative way SIFT and SP-SIFT will be used to describe both the appearance and the motion. In the second phase of our algorithm will be used the Bag of Words (BoW) model to extract a representative set of appearance and movement descriptors. From them will be calculated a histogram of appearance frequencies of these descriptors by each one of the videos. Thus we will be able to reduce the dimensionality of the problem by facilitating the classification task. In the third and final phase of the algorithm the previous histograms will be used to train a classifier capable of labeling a new video entry as violent or non-violent.

To evaluate the performance of the algorithm, the cross-validation method has been used in order to obtain reliable results. The main measure of error used is the percentage of success in the test phase. The matrices of confusion have also been extracted to know more accurately the procedence of the mistakes made.

As for the conclusions, we have that SP-SIFT gets a better performance in the task of describing the appearance of the videos, but SIFT improves in the description of the movement. The Lagrangian measures are able to improve the results obtained by the optical flow fields because they are able to take into account more than two frames. The integration parameters of these measures - number of frames that are considered for the calculation of the motion - play a fundamental role in the performance of the classifier obtaining the best results when its value is of 5-6 frames.

Our main objective was to train a classifier who was able to discriminate between violent and non-violent events. This objective has been fulfilled although we have not been able to improve the results obtained by the reference algorithm, LaSIFT.

Now, knowing the weaknesses found during the evaluation of the project, we can establish several lines of future work that could achieve performance improvements in the classification.

5.2 Future work

From the results obtained and the study carried out before the beginning of the project, we can establish different lines of future work.

First, it would be good to test the algorithm developed for the classification of other events or other simpler dataset. With this we could evaluate the results more comprehensively and extract a greater number of conclusions that would lead us to focus on the aspects to be improved. The idea would be to use Hockey Fight dataset [10] which contains videos with fewer challenges than the dataset used.

In addition to this, the number of clusters used in the Bag of Words could be adjusted in order to optimize the results.

Since five different classifiers have been trained for each of the experiments performed and then the success rate has been calculated by the arithmetic mean of each of the results of the individual experiments, one of the improvements that could lead to a significant increase Of performance would be the combination of those classifiers to draw a single conclusion. The same can be applied to the use of different classifiers trained with different values of the integration parameter. As we know, although several classifiers are based on the same characteristics to perform the classification, does not mean that they are wrong in the same points, so we could take advantage of this to make a general classifier more robust. There are different combination strategies, for example, you could use the majority voting technique or, based on the probabilities of belonging to each class, could use different rules such as the product, the maximum or average. The simplest, the majority vote, would be to evaluate the conclusion of the five classifiers and finally label the video entry with the decision that has taken the largest number of classifiers. In addition, it can be established that each classifier has a different weight in the decision, so that the classifier with the best individual performance has a greater weight in the final decision and the weight of the individual classifier with the lowest performance is lower.

Lastly, and taking into account that the results obtained when using the implementation of the global motion compensation of the camera, an exhaustive study should be carried out on the reasons why this compensation has not increased the performance. In addition, different compensation techniques could be studied and tested to see if the performance of the clasification increases. In this case, special attention should be given to the possible increases computational time of the algorithm.

Bibliography

[1] J. Nam, M. Alghoniemy, and A. H. Tewfik, "Audio-visual content-based violent scene characterization," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, pp. 353–357, IEEE, 1998.

[2] Y. Gong, W. Wang, S. Jiang, Q. Huang, and W. Gao, "Detecting violent scenes in movies by auditory and visual cues," in *Pacific-Rim Conference on Multimedia*, pp. 317–326, Springer, 2008.

[3] J. Lin and W. Wang, "Weakly-supervised violence detection in movies with audio and video based co-training," in *Pacific-Rim Conference on Multimedia*, pp. 930–935, Springer, 2009.

[4] T. Giannakopoulos, A. Makris, D. Kosmopoulos, S. Perantonis, and S. Theodoridis, "Audio-visual fusion for detecting violent scenes in videos," in *Hellenic Conference on Artificial Intelligence*, pp. 91–100, Springer, 2010.

[5] W.-H. Cheng, W.-T. Chu, and J.-L. Wu, "Semantic context detection based on hierarchical audio models," in *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 109–115, ACM, 2003.

[6] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence content classification using audio features," in *Hellenic Conference on Artificial Intelligence*, pp. 502–507, Springer, 2006.

[7] C. Clarin, J. Dionisio, M. Echavez, and P. Naval, "Dove: Detection of movie violence using motion intensity analysis on skin and blood," *PCSC*, vol. 6, pp. 150–156, 2005.

[8] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6, IEEE, 2012.

[9] O. Déniz, I. Serrano, G. Bueno, and T.-K. Kim, "Fast violence detection in video," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2, pp. 478–485, IEEE, 2014.

[10] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *International Conference on Computer Analysis of Images and Patterns*, pp. 332–339, Springer, 2011.

[11] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[12] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European conference on computer vision*, pp. 428–441, Springer, 2006.

[13] M.-y. Chen and A. Hauptmann, "Mosift: Recognizing human actions in surveillance videos," 2009.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, "Violent video detection based on mosift feature and sparse coding," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3538–3542, IEEE, 2014.

[16] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[17] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, 2013.

[18] T. Senst, V. Eiselein, and T. Sikora, "A local feature based on lagrangian measures for violent video classification," in *6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15)*, pp. 1–6, IET, 2015.

[19] C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conference*, vol. 15, pp. 10–5244, Citeseer, 1988.

[20] C.-F. Tsai, "Bag-of-words representation in image annotation: A review," *ISRN Artificial Intelligence*, vol. 2012, 2012.

[21] F. Navarro, M. Escudero-Vinolo, and J. Bescós, "Sp-sift: enhancing sift discrimination via super-pixel-based foreground-background segregation," *Electronics Letters*, vol. 50, no. 4, pp. 272–274, 2014.

[22] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," *Computer vision–ECCV 2008*, pp. 705–718, 2008.

[23] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1385–1392, 2013.

[24] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[25] A. Kuhn, T. Senst, I. Keller, T. Sikora, and H. Theisel, "A lagrangian framework for video analytics," in *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on*, pp. 387–392, IEEE, 2012.

[26] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

Appendix A

Motion representation

The following representation is used to represent the motion estimation of the scene.

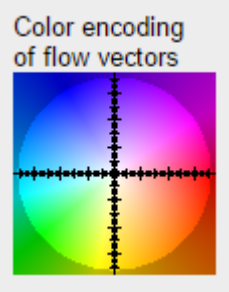


Figure A.1: Color encoding motion components.

As we see in Figure A.1 the different shades indicate different positions of the extracted movement vector, so that a movement with (-1;-2) coordinates will have greenish tones and movement whose components are. for example, (-2, 1) will be represented in bluish colors.

Appendix B

Confusion matrices

B.1 Appearance description: SIFT vs SP-SIFT

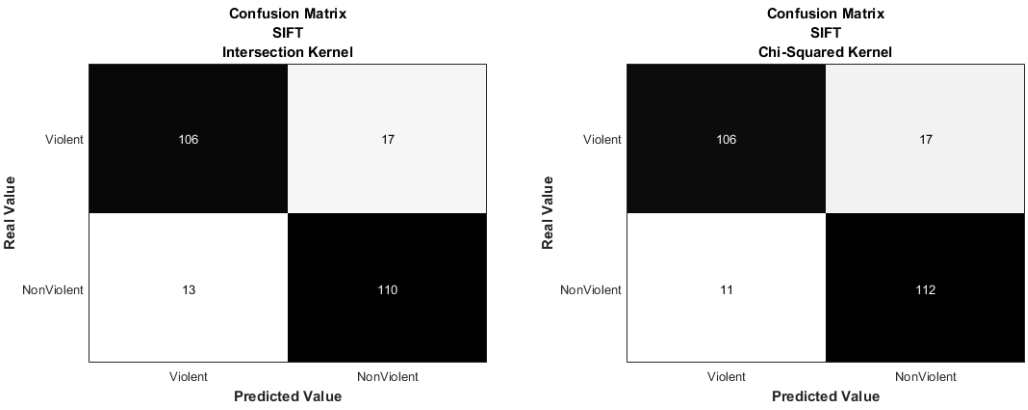


Figure B.1: Appearance description: SIFT.

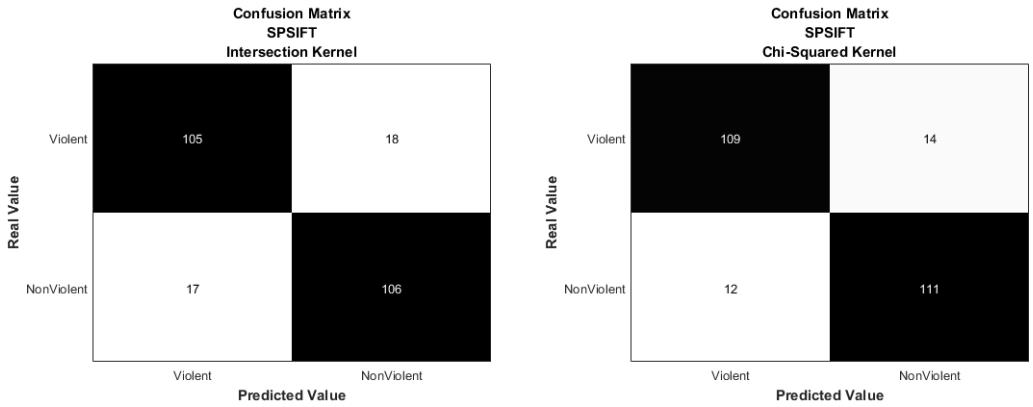


Figure B.2: Appearance description: SP-SIFT.

B.2 Deep Flow description: SIFT vs SP-SIFT

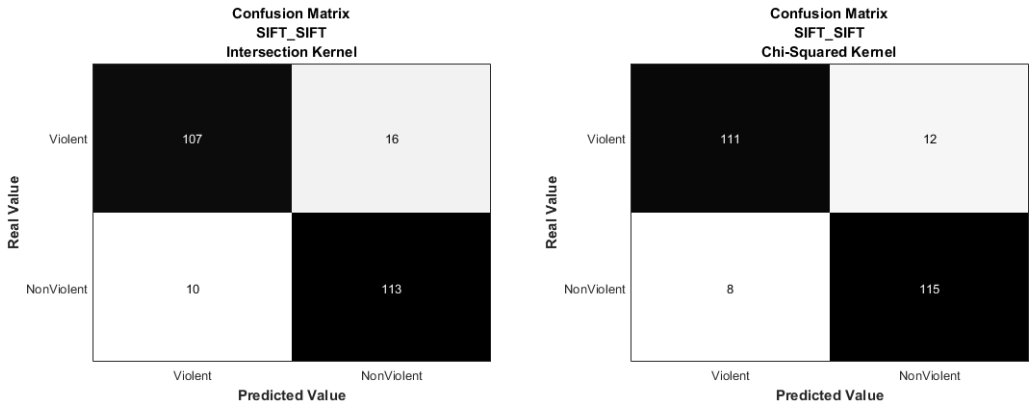


Figure B.3: Appearance description: SIFT. Deep flow: SIFT.

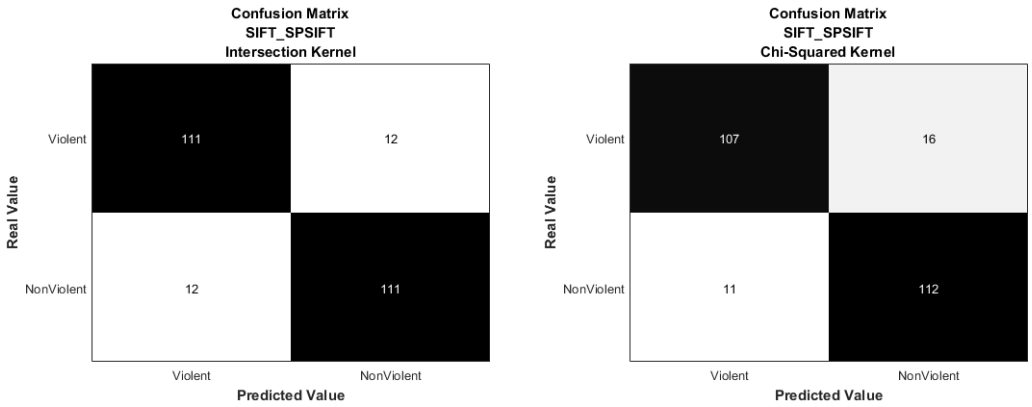


Figure B.4: Appearance description: SIFT. Deep flow: SP-SIFT.

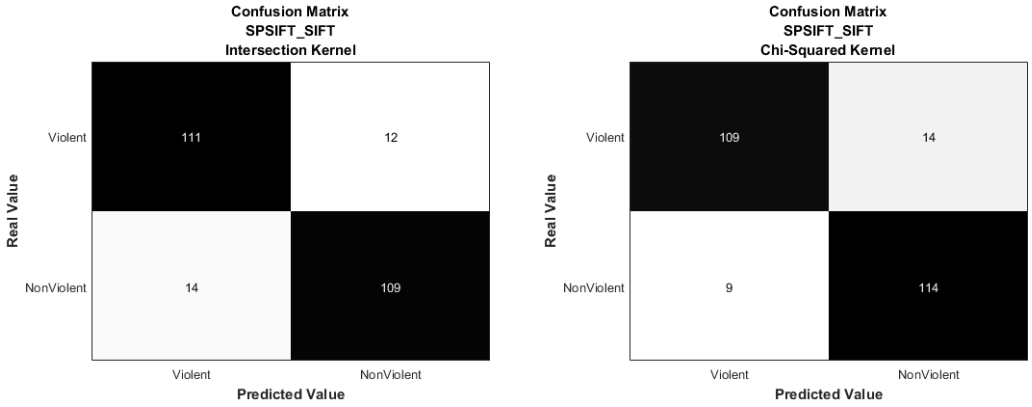


Figure B.5: Appearance description: SP-SIFT. Deep flow: SIFT.

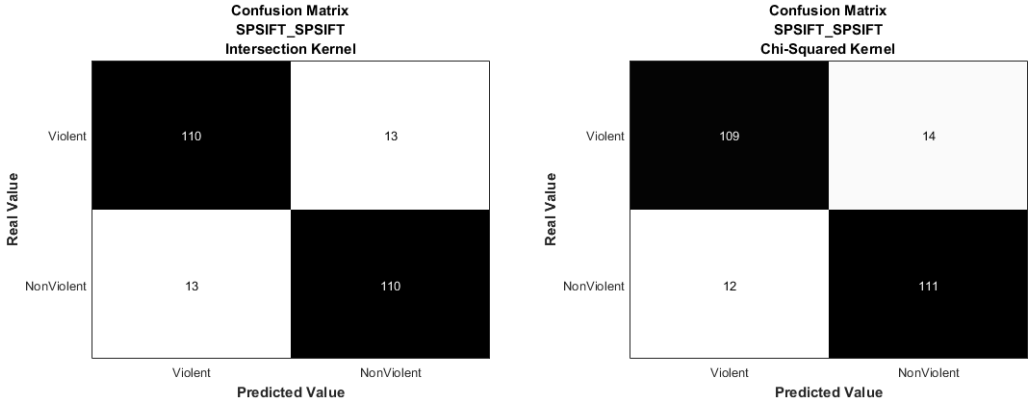


Figure B.6: Appearance description: SP-SIFT. Deep flow: SP-SIFT.

B.3 Direction Lagrangian measures description: SIFT vs SP-SIFT

B.3.1 LaSIFT

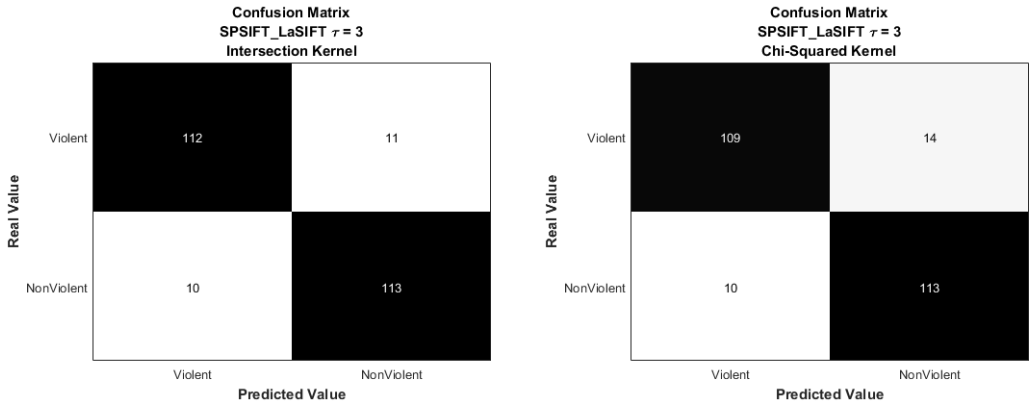


Figure B.7: Appearance description: SP-SIFT. Lagrangian $\tau = 3$: SIFT.

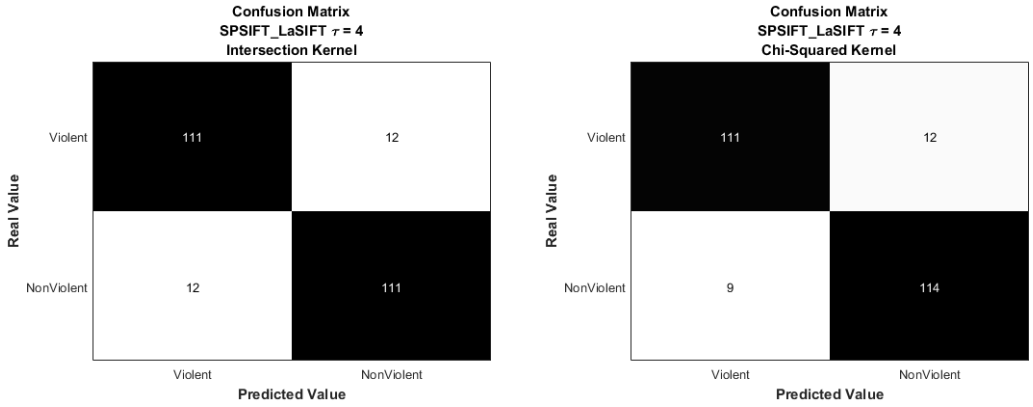


Figure B.8: Appearance description: SP-SIFT. Lagrangian $\tau = 4$: SIFT.

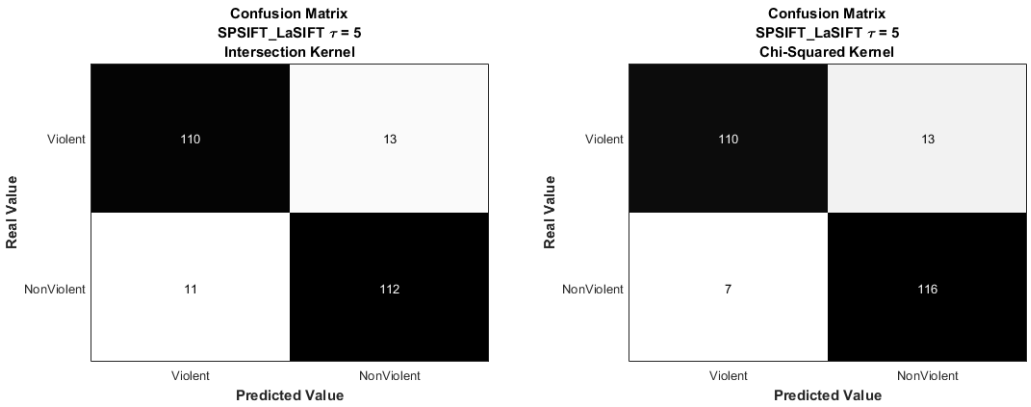


Figure B.9: Appearance description: SP-SIFT. Lagrangian $\tau = 5$: SIFT.

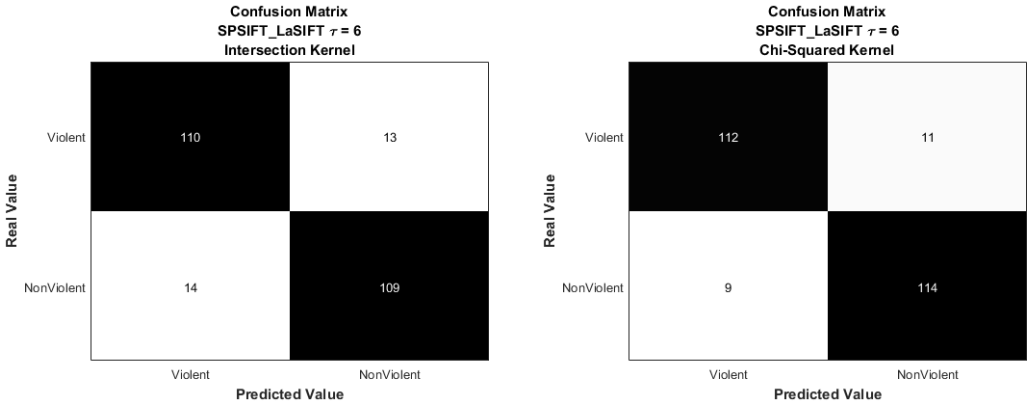


Figure B.10: Appearance description: SP-SIFT. Lagrangian $\tau = 6$: SIFT.

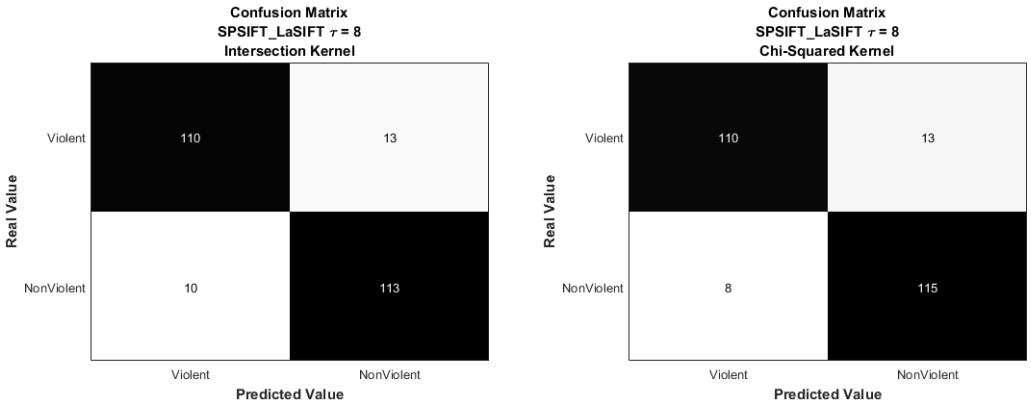


Figure B.11: Appearance description: SP-SIFT. Lagrangian $\tau = 8$: SIFT.

B.3.2 LaSP-SIFT

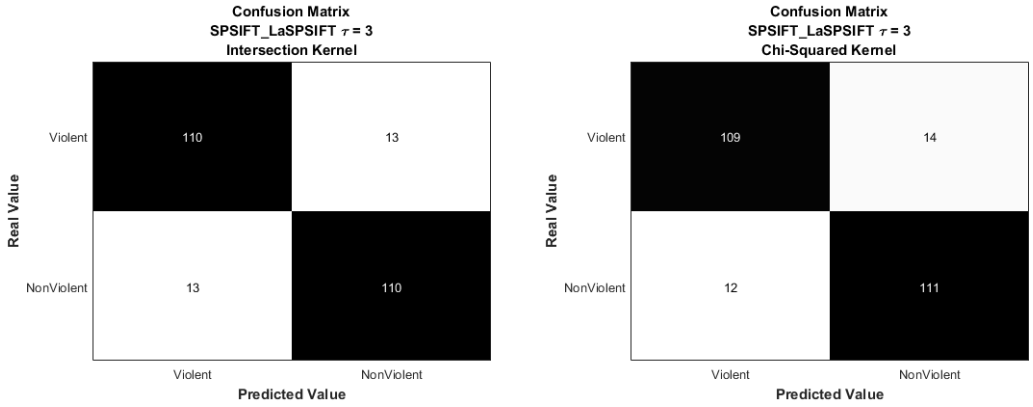


Figure B.12: Appearance description: SP-SIFT. Lagrangian $\tau = 3$: SP-SIFT.

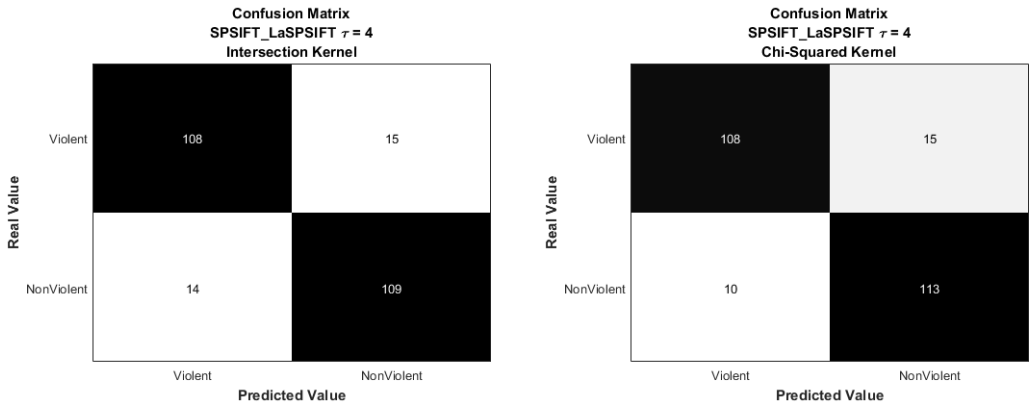


Figure B.13: Appearance description: SP-SIFT. Lagrangian $\tau = 4$: SP-SIFT.

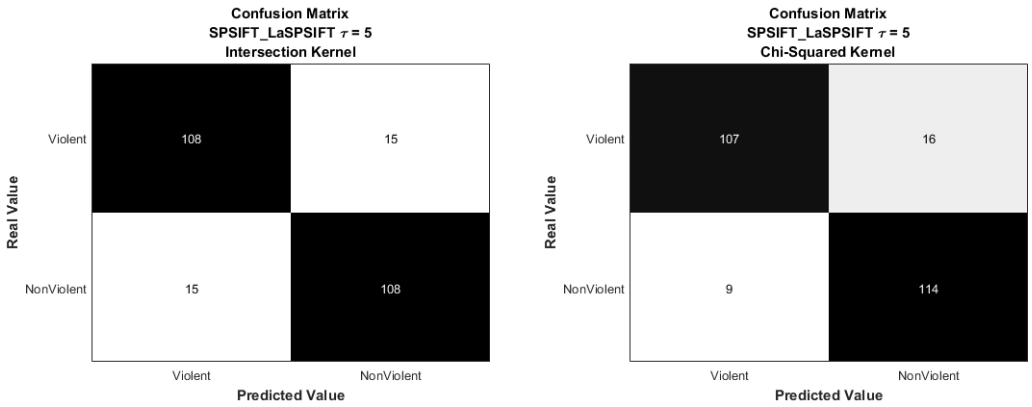


Figure B.14: Appearance description: SP-SIFT. Lagrangian $\tau = 5$: SP-SIFT.

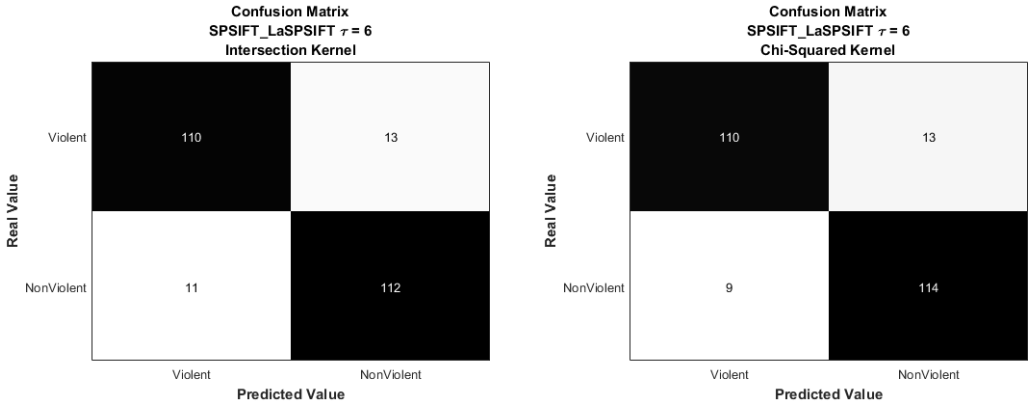


Figure B.15: Appearance description: SP-SIFT. Lagrangian $\tau = 6$: SP-SIFT.

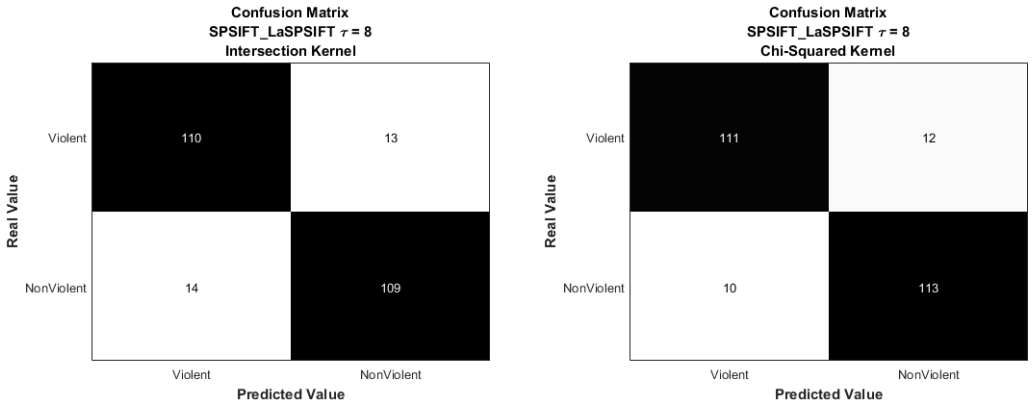


Figure B.16: Appearance description: SP-SIFT. Lagrangian $\tau = 8$: SP-SIFT.

B.4 Influence of global motion compensation

B.4.1 LaSIFT with global motion compensation

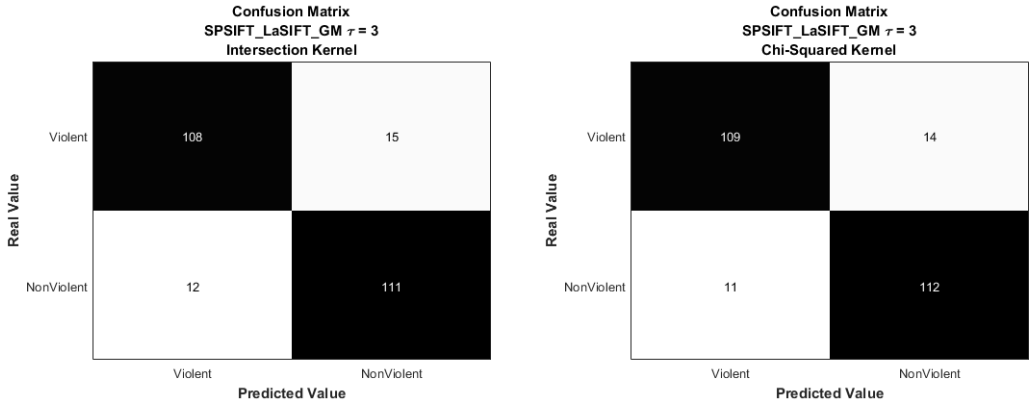


Figure B.17: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 3$: SIFT.

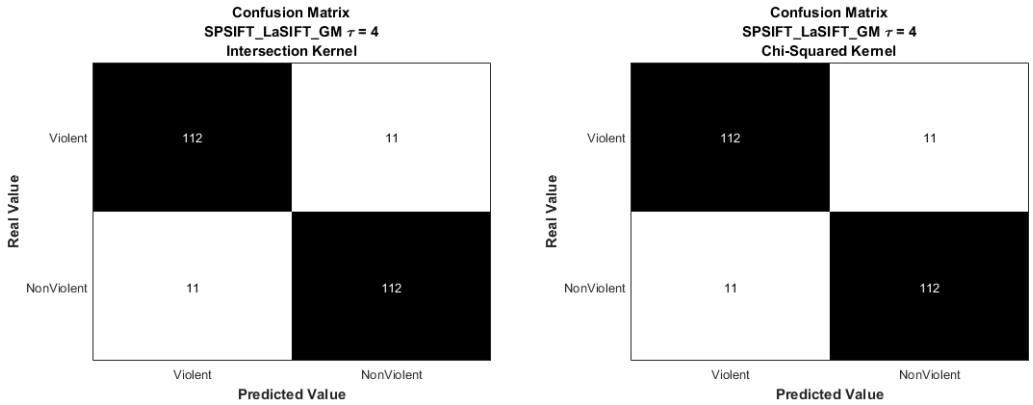


Figure B.18: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 4$: SIFT.

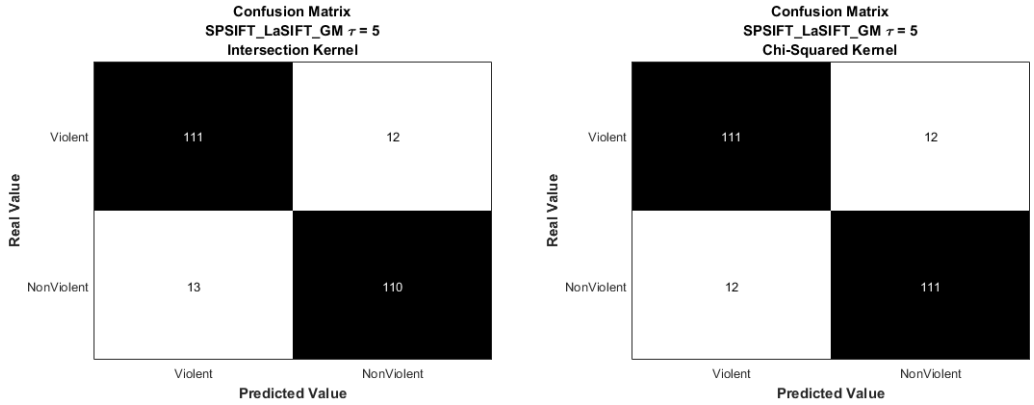


Figure B.19: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 5$: SIFT.

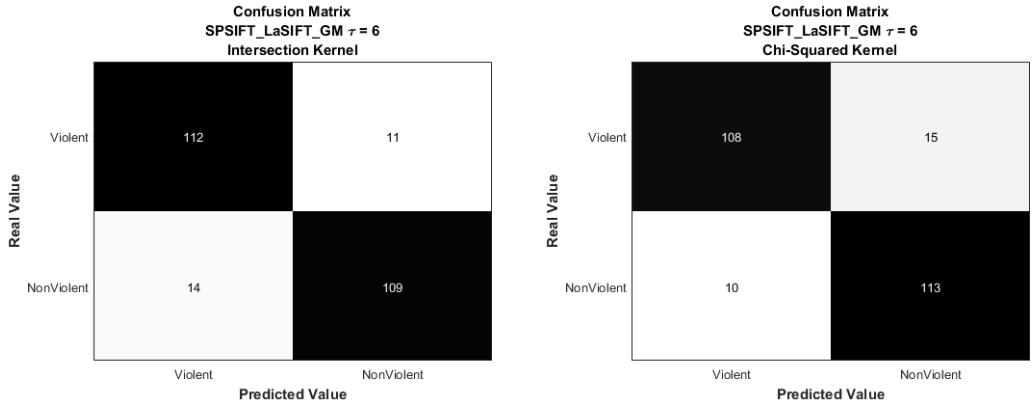


Figure B.20: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 6$: SIFT.

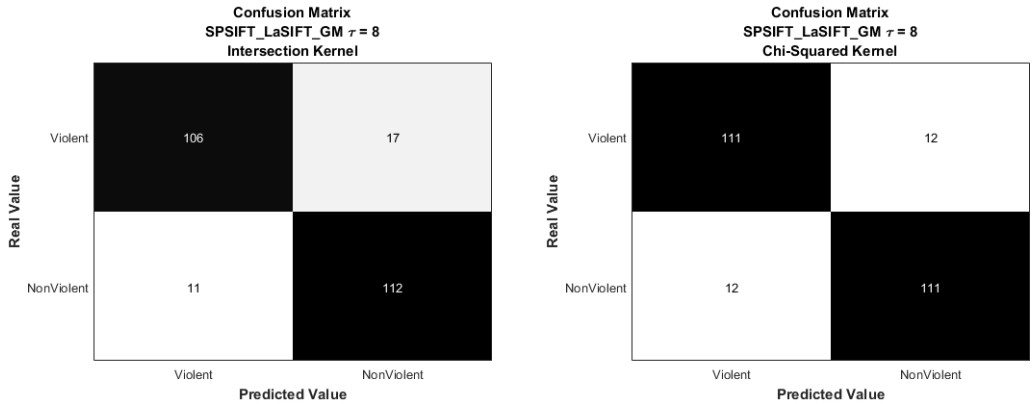


Figure B.21: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 8$: SIFT.

B.4.2 LaSP-SIFT with global motion compensation

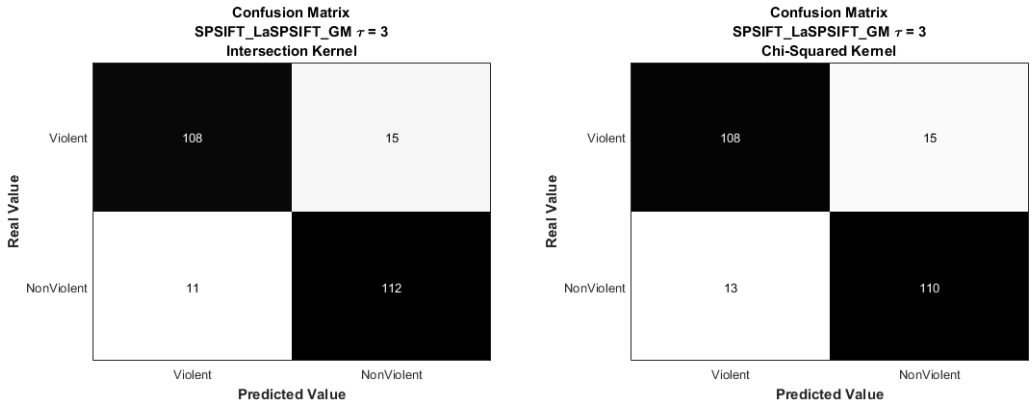


Figure B.22: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 3$: SP-SIFT.

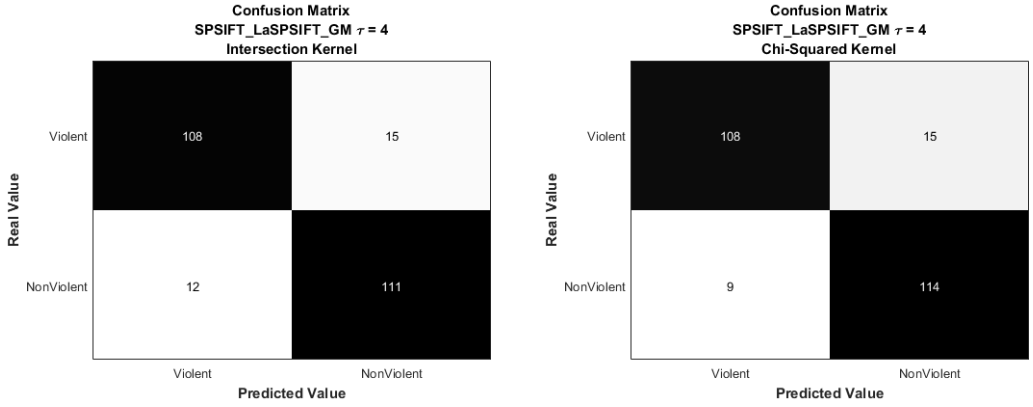


Figure B.23: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 4$: SP-SIFT.

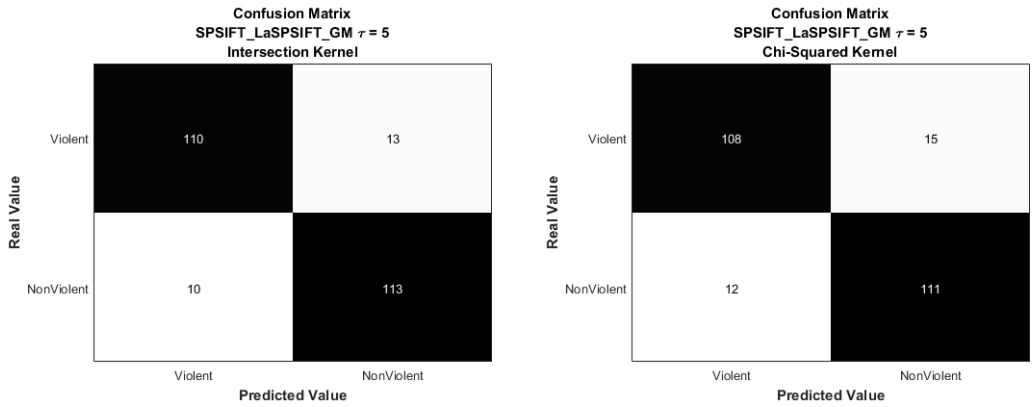


Figure B.24: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 5$: SP-SIFT.

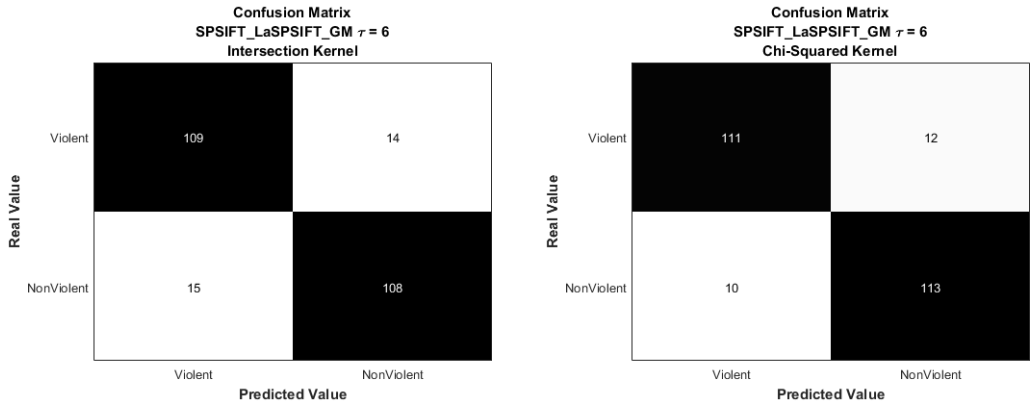


Figure B.25: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 6$: SP-SIFT.

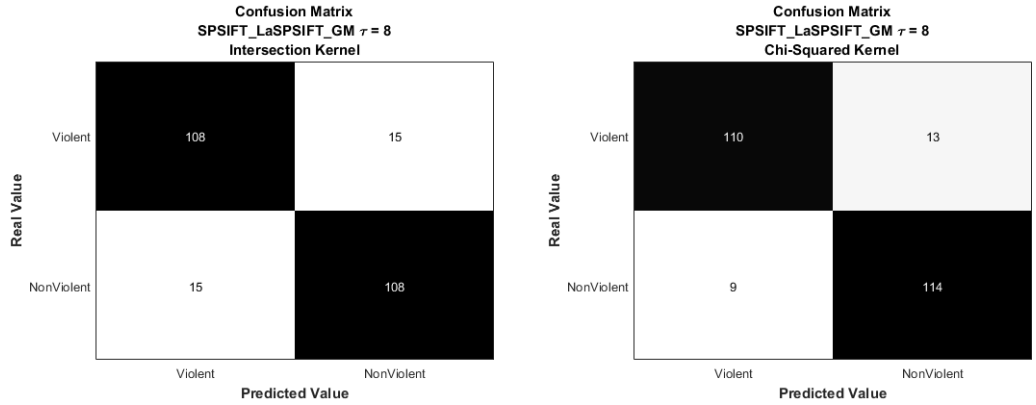


Figure B.26: Appearance description: SP-SIFT. Lagrangian with GM $\tau = 8$: SP-SIFT.